UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior - Leganés

INGENIERÍA DE TELECOMUNICACIÓN

Departamento de Teoría de la Señal y Comunicaciones

PROYECTO FIN DE CARRERA

MÉTODOS DE REDUCCIÓN DE LA CARGA COMPUTACIONAL DE CLASIFICADORES MULTICLASE BASADOS EN MÁQUINAS DE VECTORES SOPORTE

AUTOR: SARA GONELL SÁNCHEZ-SECO
TUTOR: Dr. EMILIO PARRADO HERNÁNDEZ

Leganés, 2010

Agradecimientos.

Este proyecto supone la etapa final de un largo y duro camino en el que he invertido el esfuerzo de los últimos años. Llegado este momento tan ansiado y antes de terminar, debo agradecer a todos los que me han ayudado con sus ánimos a llegar hasta aquí.

A mis padres, para quienes no existen suficientes palabras para agradecer todo lo que han hecho por mí. Gracias por su amor, su infinito apoyo, sus esfuerzos y abnegación absoluta y sobre todo por haberse sentido siempre los padres más orgullosos y felices viendo los logros de sus hijas.

A mi hermana Esther, por ser mi modelo a seguir durante toda mi vida. Admiro tu forma de enfrentarte a la vida, tu entereza y tu gran personalidad. Gracias por abrirme siempre todos los caminos y facilitarme las cosas y, sobre todo, por confiar en mí.

A Felipe, por compartir tantos buenos momentos en estos últimos años, siendo mi mejor amigo y llegando a conocerme casi más que a ti mismo. Gracias por ser mi compañero de vida y por darme todo tu amor y quererme tanto. Te quiero mucho.

A todos los buenos amigos que me llevo de la universidad, a los que se quedaron y a los que decidieron emprender otros caminos. Por todas las cosas que hemos vivido juntos, dentro y fuera, sois una de las mejores cosas que me he encontrado estos años. Espero que el final de esta etapa suponga el principio de otra mejor.

A Belén, mi amiga más “abuela”, porque tus ganas de superarte cada vez más me han dado la suficiente energía y confianza para conseguir llegar hasta aquí, ahora sólo nos queda disfrutar.

A mi familia en general, en especial a los que ya no están, porque nunca han dudado de mí y sabían, mejor que yo misma, que podría con todo aquello que me propusiera.

A mi tutor Emilio, por introducirme en el campo del aprendizaje automático y por lo mucho que he aprendido trabajando con él.

A aquellos que faltan en esta página y que también han colaborado en que en este momento yo sea tal y como soy.

A todos gracias.

Sara

Resumen.

En los últimos años se ha experimentado un incremento exponencial de la información que se espera que continúe creciendo en el futuro. Por este motivo es necesaria la organización por medios automáticos de toda esta información para facilitar el acceso, la búsqueda y el análisis de la misma.

El aprendizaje automático se encarga de diseñar y desarrollar algoritmos que permitan a los ordenadores ser más eficientes y realizar tareas sin apenas supervisión humana. Este tipo de aprendizaje tratará de producir de manera automática modelos, reglas o patrones a partir de una serie de datos iniciales. El aprendizaje automático está por tanto íntimamente relacionado con campos tan extensos como pueden ser la minería de datos, la estadística o el reconocimiento de patrones, entre otros. En las últimas décadas, dada la gran demanda de estas aplicaciones, se ha visto incrementado de manera notable el desarrollo de nuevas técnicas de aprendizaje automático.

En este Proyecto de Fin de Carrera se hará una introducción a los diferentes tipos de aprendizaje automático así como a su aplicación a diversos problemas de clasificación, sobre todo, en entornos multiclase. De entre éstos, se hará especial hincapié en los problemas de clasificación de textos e imágenes de dígitos manuscritos en los que se aplicará una técnica de aprendizaje automático supervisada. Este tipo de técnicas de aprendizaje se refieren a todas aquellas aplicaciones o procesos en los que se dispone de información como son los valores de entrada del sistema y los valores de salida deseados.

Uno de los objetivos principales es utilizar la técnica de aprendizaje supervisado basada en máquinas de vectores soporte para realizar varias aproximaciones a problemas de multiclasificación con el fin de resolver algunas desventajas que presentan los algoritmos de combinación de clasificadores tradicionales como pueden ser la complejidad de estos métodos de clasificación así como la elevada carga computacional y temporal en la evaluación de los resultados.

En este Proyecto de Fin de Carrera se explican detalladamente las dos aproximaciones propuestas para problemas multiclase en las que se aplicarán diversas estrategias de combinación de clasificadores. Por último, se realizará un estudio comparativo de estos algoritmos y de esta manera poder comprender mejor las características cualitativas y cuantitativas de éstos.

Abstract.

In recent years there has been an exponential increase of information that is expected to continue growing in the future. Therefore, it is necessary to organize all this information by automatic means to facilitate access, search and analysis of that information.

Machine learning is responsible for designing and developing algorithms that allow computers to be more efficient and perform tasks without human supervision. This type of learning will automatically produce models, rules or patterns from a series of initial data. Therefore, machine learning is closely related to fields such as data mining, statistics or pattern recognition, among others. During the last decades, due the huge demand in these applications, the development of new machine learning techniques there has been increased.

This Final Project makes a brief introduction to different types of machine learning techniques and its application to various classification problems, especially in multi-class environments. Among these, special interest will be paid to classification problems of texts or images of handwritten digits in which we apply a supervised machine learning technique. This type of learning techniques refers to all those applications or processes in which some information is available as the input values of the system and the desired output values.

One of the main goals is to use a supervised learning technique based on support vector machines to perform several approaches to multi-class problems in order to try to solve some disadvantages that traditional combination of classifiers algorithms have. Some of these disadvantages can be the complexity of these classification methods and a huge temporal and computational cost in the evaluation of the results.

This Final Project explains in detail the two approaches that have been proposed for multi-class problems in which we will apply various strategies for combining of classifiers. Finally, we will perform a comparison of those algorithms and so as to comprehend easily the qualitative and quantitative features of these.

Índice General.

Índice de Figuras.

Índice de Tablas.

Capítulo 1

Introducción.

En este primer capítulo del proyecto vamos a introducir el marco tecnológico en el que se van a cimentar nuestras investigaciones. Para empezar se va a presentar un resumen de las diferentes técnicas de aprendizaje automático e introduciremos problemas de clasificación binaria y multiclase. Posteriormente, nos centraremos en los problemas de clasificación de textos y de imágenes de números manuscritos y finalmente marcaremos los objetivos que queremos conseguir.

El objetivo del proyecto es la reducción del coste y la carga computacional en el ámbito de la clasificación automática para el caso de problemas con múltiples categorías tanto de textos como de imágenes de dígitos manuscritos. Para ello utilizaremos una de las técnicas más utilizadas en la clasificación automática denominada Máquina de Vectores Soporte o SVM *(Support Vector Machines)*.

1.1. Marco Tecnológico

1.1.1. Aprendizaje Automático o Máquina

Desde épocas antiguas han existido métodos de recopilación de información, usualmente de manera escrita, como la creación de documentos, libros, etc. En los últimos años ha habido un incremento exponencial de información, sobre todo en formato digital, y se espera que continúe creciendo en el futuro.

Estas expectativas hacen que sea necesaria la organización por medios automáticos de todos estos documentos y contenidos digitales para facilitar el acceso, la búsqueda y el análisis de la información. No obstante, el problema de organización es complejo y por tanto se realizan continuamente investigaciones para encontrar métodos apropiados y para mejorarlos. Por ejemplo, en el tema del tratamiento automático de textos se han ido creando varias líneas de investigación en las que se puede encontrar la clasificación de los diversos textos y la búsqueda, recuperación y extracción de la información.

Inicialmente, la organización de la información se realizaba de manera manual y por lo tanto el proceso era muy difícil, costoso y requería mucho tiempo. Para intentar solucionar estos problemas, en los últimos años, surge una de las principales áreas de investigación para desarrollar nuevas técnicas de aprendizaje y clasificación de forma automática.

El aprendizaje automático se basa en la obtención de determinadas características de un ejemplo particular con el fin de poder ser clasificado dentro de una categoría, utilizando para ello una colección o conjunto de datos inicial. Una vez que se obtienen las características principales de cada clase se puede diseñar un clasificador que permita catalogar nuevos ejemplos no entrenados previamente.

El aprendizaje automático permite una gran variedad de enfoques y técnicas, y puede aplicarse a diferentes problemas de clasificación. Existen tres tipos de enfoques de aprendizaje dependiendo de la base de conocimiento: supervisado, no supervisado y semi-supervisado o parcialmente supervisado.

1.1.2. Tipos de Aprendizaje Automático

1.1.2.1. Aprendizaje Supervisado

Los algoritmos basados en aprendizaje supervisado, [Theodoridis y. Koutroumbas, 1998], son aquellos que utilizan la información suministrada por un conjunto de ejemplos de entrada etiquetados así como de las salidas deseadas. Cada uno de estos ejemplos posee información acerca de las principales características o atributos de la clase o categoría a la que representan. Este conjunto de ejemplos con etiquetas conocidas, llamado conjunto de entrenamiento, guía el proceso de aprendizaje haciendo que se obtenga un modelo de predicción/clasificación en una primera fase de entrenamiento. En la fase de test, se prueba el modelo con nuevos datos introducidos que no han sido vistos anteriormente cuya etiqueta es desconocida y se realiza una predicción de la clase o categoría.

Los algoritmos supervisados que se utilizan más comúnmente entre otros son el agrupamiento k-NN o de los k vecinos más próximos ([Fix y Hodges, 1951]), clasificador Naive Bayes ([Duda y Hart, 1973]), máquinas de vector soporte (SVM, [Cristianini y Shawe-Taylor, 2000]), etc. Estos dos últimos métodos están especialmente indicados en el área de clasificación de textos.

1.1.2.2. Aprendizaje No Supervisado

En los algoritmos basados en aprendizaje no supervisado, al contrario que el aprendizaje supervisado, no se posee información sobre la salida deseada, por ejemplo, en el caso de la clasificación no se conoce la clase o categoría a la que pertenece el patrón. Este tipo de algoritmos se basan en el descubrimiento de algunos patrones de semejanza entre los datos que permitan la separación de los ejemplos en las distintas clases. De este modo las categorías se pueden caracterizar extrayendo diferentes parámetros, características, relaciones, etc.

Los algoritmos no-supervisados más comunes son Cobweb ([Fisher, 1987]), algoritmo EM (Expectation Maximization, [Dempster er al., 1977]), k-Means ([MacQueen, 1967]), etc. Éste último es uno de los más utilizados en el área de agrupamiento de textos.

1.1.2.3. Aprendizaje Semi-Supervisado

En la década de los 90 aparecen diversos trabajos de varios autores como [Castelli y Cover, 1995], [Bensaid et al., 1996] y [Blum y Chawla, 2001] en los que se presenta un nuevo concepto de aprendizaje, el aprendizaje semi-supervisado o, también llamado en la literatura, aprendizaje parcialmente supervisado.

El aprendizaje semi-supervisado es una combinación de los dos tipos de aprendizaje citados anteriormente: supervisado y no supervisado. Este tipo de algoritmos utiliza tanto un conjunto de datos etiquetados asociados a una determinada clase como aquellos datos no etiquetados ni asociados a una categoría para generar un modelo de predicción/clasificación adecuado. La idea en la que se basan estos algoritmos es entrenar y aprender una máquina con los datos que están asociados a una clase y posteriormente asignar una categoría a los datos que no la tengan asociada basándose en técnicas de agrupamiento.

Normalmente se asume que los dos tipos de conjuntos, sin etiqueta y etiquetados, provienen de la misma distribución y que la cantidad de datos etiquetados es mucho menor que la de los datos sin etiquetar.

Las principales ventajas de estos algoritmos es que requieren menor esfuerzo humano ya que no es necesario etiquetar todo el conjunto de datos y dan una mayor exactitud con respecto de los otros dos tipos de aprendizajes.

Algunos de los algoritmos semi-supervisados más comunes son los basados en EM como Co-EM [Nigam y Ghani, 2000], Bootstrapping [Efron y Tibshirani, 1993], Self-training [Nigam y Ghani, 2000)], Co-training [Blum y Mitchell, 1998], máquinas de soporte vectorial transductivo o semi-supervisado (T-SVM o S^3VM) (pueden verse varios ejemplos en [Bennett y Demiriz, 1998] o en [Joachims, 2001]), etc.

1.2. Introducción a los Problemas de Clasificación

A lo largo de la historia, los problemas propuestos a resolver por los métodos de automatización se han centrado principalmente en los llamados “problemas de clasificación”. El término clasificación se ha utilizado para denotar el reconocimiento y agrupación de objetos que pueden estar representados de distinta forma dentro de un determinado sistema. Dichos objetos suelen estar definidos a través de un conjunto de clases o categorías, sobre las cuales se debe hacer un previo aprendizaje para proceder a su posterior reconocimiento.

En un principio podríamos pensar que estos sistemas podrían moverse en un espacio con un conjunto binario de clases con dos categorías a distinguir. De aquí surge toda la teoría y estudio de la clasificación binaria. A simple vista podemos ver que se necesitaría alguna herramienta para trabajar con una clasificación que no fuese simplemente binaria, ya que el mundo tiene inherentemente un funcionamiento multiclase. De esta manera pasaríamos a clasificar un objeto entre *N* posibles clases definidas en nuestro sistema.

1.2.1. Problemas de Multiclasificación

A partir de la década de los 90 surge una nueva metodología de clasificación que, a diferencia de las técnicas tradicionales que se basan en clasificar nuevos patrones utilizando un único clasificador, tienen en cuenta todas las decisiones aportadas por múltiples clasificadores. Estas nuevas técnicas tienen en cuenta todas las hipótesis válidas aportadas por cada uno de los clasificadores realizando la combinación de las predicciones del conjunto para obtener una clasificación final.

El uso de los multiclasificadores ha aumentado en los últimos años ya que resuelven los problemas de sobreadaptación u *overfitting* y es posible obtener buenos resultados con conjuntos de entrenamiento de pocas muestras. También la idea de la combinación de diferentes clasificadores se realiza con el principal propósito de mejorar las prestaciones de la clasificación con un clasificador individual. Asimismo, los multiclasificadores pueden servir para descomponer un problema complejo en múltiples subproblemas que sean más sencillos de entender y resolver.

Algunos aspectos de la clasificación con un único clasificador que hacen que estos sistemas con múltiples clasificadores puedan ser mejores son:

- La decisión combinada mejora a las decisiones individuales
- Los errores correlacionados de los clasificadores individuales pueden ser eliminados por medio de la combinación, considerándose el total de las decisiones
- Los patrones de entrenamiento pueden no proporcionar la información suficiente para seleccionar el mejor clasificador
- El algoritmo de aprendizaje puede no ser adaptado para solucionar el problema
- El espacio individual de búsqueda puede no contener la función que se propone

Estas razones pueden ser analizadas más detalladamente desde un punto de vista estadístico, computacional y representacional:

- *Estadística*: Un algoritmo de aprendizaje puede verse como la búsqueda de la mejor decisión en un espacio de hipótesis. El problema estadístico se presenta cuando el tamaño del espacio de decisiones es demasiado grande para la cantidad de datos de entrenamiento disponibles. Pueden existir varias hipótesis que logran la misma exactitud para los datos de entrenamiento y el algoritmo de aprendizaje debe elegir sólo una de éstas. Entonces, puede existir el riesgo de que la hipótesis o decisión tomada no prediga correctamente los patrones nuevos que se presenten en el futuro.

- *Computacional*: Muchos algoritmos de aprendizaje intentan buscar un mínimo global para el problema que se presenta. En los casos en los que el conjunto de datos de entrenamiento es muy grande el problema de encontrar la mejor solución puede resultar muy difícil de resolver computacionalmente. La combinación de varios clasificadores individuales construidos para realizar la búsqueda local puede proporcionar una mejor aproximación a la decisión correcta desconocida que cualquiera de los clasificadores individuales.

- *Representacional*: En la mayoría de aplicaciones de aprendizaje automático la decisión correcta no puede ser representada por ninguna de las decisiones tomadas por los clasificadores individuales. En algunos casos una suma de pesos de las decisiones individuales amplía el espacio de las funciones que pueden ser representadas. La utilización de un sistema de multiclasificación en el que se ponderan las decisiones individuales puede ser capaz de formar una aproximación más exacta a la decisión correcta.

Existen diversos sistemas de clasificación múltiple dependiendo de sus características. Entre ellas podemos citar: el número de clasificadores individuales que se van a combinar, el tipo de clasificador utilizado (vecinos más cercanos, redes neuronales, SVM, etc), el tamaño de los conjuntos de entrenamiento utilizados por los clasificadores así como sus características y la técnica de combinación o agregación de las decisiones individuales (voto mayoritario simple o ponderado, reglas estadísticas, etc).

Actualmente existen muchos ámbitos de aplicación en los que se puede emplear la multiclasificación como la clasificación de textos, el reconocimiento de imágenes, clasificación de páginas web, entre otros muchos.

Por estos motivos, las investigaciones acerca de la clasificación con múltiples clasificadores están en pleno auge en estos días.

1.3. Ejemplos de Problemas de Clasificación

1.3.1. Clasificación Automática de Textos

La clasificación de textos es la tarea de clasificar automáticamente un conjunto de documentos que se encuentran dentro de una colección predefinida en categorías o temas.

Desde principios de la década de los 90 se ha avanzado notablemente en el estudio de los sistemas de clasificación de texto que se basan el la combinación de varias tecnologías como el aprendizaje automático y la recuperación de la información.

Este enfoque se ha convertido en el dominante para construir sistemas de clasificación de textos y la idea básica es realizar un proceso que construya un clasificador que observe las características de un conjunto de documentos que previamente han sido clasificados. De esta manera el problema de la clasificación de textos se convierte en un problema se aprendizaje automático supervisado presentado en la sección 1.1.2.1.

Es necesario medir la efectividad y las prestaciones de los sistemas de clasificación de textos. Para ello se suelen utilizar generalmente medidas de recuperación de la información como son la precisión, la cobertura y la medida F, entre otras. Las más utilizadas se presentarán y describirán más adelante en el capítulo 4 en la sección 4.3, donde se determinará qué medidas de evaluación usaremos para comparar y valorar las prestaciones y el buen rendimiento de los diferentes métodos experimentales.

1.3.2. Clasificación Automática de Imágenes

La clasificación o reconocimiento de caracteres tiene como objetivo asociar una imagen a la clase correspondiente de entre un conjunto de símbolos que componen un determinado alfabeto. Este mecanismo puede aplicarse en varios dominios o situaciones como el reconocimiento de letras o dígitos de manera aislada o incluso el análisis o comprensión de documentos.

La complejidad del sistema de reconocimiento puede distinguirse según el tipo de caracteres que se van a analizar. De esta manera se pueden diferenciar varios tipos de análisis como el reconocimiento de uno o varios tipos de letras impresas, de cualquier letra impresa, de caracteres manuscritos, etc.

El reconocimiento de caracteres se suele usar en aplicaciones del mundo real para automatizar la lectura de direcciones postales, cheques bancarios, varios tipos de formularios y lectores de texto para discapacitados, entre otros. El objetivo o fin de estos sistemas de clasificación automática es poder proporcionar una alternativa fiable basada en el reconocimiento de caracteres escritos manualmente por el usuario y que no tenga un coste computacional demasiado elevado. La idea fundamental es que si se dispone de una amplia base de datos de caracteres escritos por diferentes personas ésta se puede utilizar para mejorar la fiabilidad de la clasificación.

En este proyecto se van a estudiar varias estrategias de clasificación aplicadas al reconocimiento de dígitos manuscritos aislados. Vamos a utilizar una base de datos de imágenes de números manuscritos para automatizar la lectura de las direcciones de los sobres del servicio postal de Estados Unidos.

1.4. Objetivos

En los últimos años, las máquinas de vectores de soporte (SVM), en vista de los buenos resultados obtenidos en diversas investigaciones, se perfilan como una buena solución para los problemas de clasificación automática. Sin embargo, la naturaleza dicotómica o binaria de este algoritmo de clasificación hace necesario e interesante el estudio de su aplicación a problemas multiclase como, por ejemplo, la clasificación de textos.

En este proyecto se van a presentar y comparar diferentes aproximaciones a los problemas de clasificación multiclase para la técnica de aprendizaje supervisada basada en máquinas de vectores soporte. Se centra en el estudio de dos aproximaciones basadas en la combinación de clasificadores SVM pareados: una deconstructiva basada en poda de clasificadores y otra constructiva. Así mismo se analizarán diversas estrategias de combinación con las que se va a predecir la clase o categoría de las muestras del conjunto de prueba del problema.

El objetivo principal del proyecto y del estudio de dichas aproximaciones a problemas multiclase es el de resolver alguna de las desventajas que presentan los algoritmos de combinación de clasificadores pareados tradicionales como pueden ser la elevada carga computacional y temporal en las etapas de la clasificación (entrenamiento y predicción), así como en la evaluación de los resultados.

En este proyecto, se van a realizar varios experimentos que tienen como fin la evaluación la bondad de las aproximaciones propuestas atendiendo a la eficacia o la precisión en la clasificación y a la consecución del objetivo fijado: la reducción del coste computacional.

1.5. Organización del Proyecto

El proyecto está estructurado de la siguiente manera. En el siguiente Capítulo presentamos el punto de partida del proyecto en el que hace una revisión de la clasificación basada en máquinas de vectores soporte y su extensión a clasificación multiclase. Por último se hace un resumen del artículo publicado por [Ye y Shang-Teng, 2007] en el que basaremos nuestras investigaciones y en el que se habla de la reducción del número de clasificadores pareados utilizando estrategias de muestreo de incertidumbre y en el que nos hemos basado para elaborar nuestras investigaciones.

En el Capítulo 3 presentamos la propuesta de diversas estrategias de combinación o fusión de clasificadores binarios utilizadas para la toma de decisiones o predicción de la clase y que son usadas para mejorar el desempeño y las prestaciones que se obtendrían únicamente con los clasificadores individuales. Por último en esta sección se explicarán las dos aproximaciones para la clasificación multiclase basada en SVM, deconstructiva y

constructiva, que permiten la reducción de la complejidad de los métodos de clasificación tradicionales.

El Capítulo 4 se centra en la presentación de los resultados obtenidos en nuestro trabajo experimental. En un principio se presentan las colecciones de datos que se van a utilizar en los experimentos, una de textos y otra de imágenes, y también se muestra el procesado de las colecciones necesario previamente. También se podrán ver las medidas de evaluación que se utilizarán para evaluar el rendimiento de cada uno de los métodos experimentales. Por último, se realiza una presentación de los resultados obtenidos al aplicar las aproximaciones propuestas en este proyecto y un análisis de los resultados obtenidos, analizando y comparando las diferentes técnicas de multiclasificación.

Finalmente, en el Capítulo 5 se concluye con las ideas extraídas de la realización de este proyecto y la explicación de posibles líneas que quedan abiertas para experimentaciones futuras.

Capítulo 2

Punto de Partida.

En este capítulo se presenta el punto de partida de nuestro trabajo. Para comenzar se describe el tipo de clasificación que vamos a utilizar para nuestros experimentos, clasificación por máquinas de vectores soporte o SVM. Posteriormente, ya que trabajaremos con colecciones de datos con múltiples categorías necesitamos describir el entorno multiclase para la clasificación SVM. Más tarde se hará una pequeña introducción a las bases de datos de textos y a sus métodos de representación. Finalmente, se va a hacer una breve presentación del artículo en el que vamos a basar nuestra investigación y experimentos

2.1. Clasificación con Máquinas de Vectores Soporte

Las máquinas de vector soporte es una de las técnicas de clasificación más utilizadas en la última década. Fue desarrollada por [Cortes y Vapnik, 1995] y más tarde fue Joachims el que las utilizó por primera vez en el área de clasificación automática en [Joachims, 1998] y en [Joachims, 2001]. Se ha demostrado que da buenos resultados en diversas aplicaciones de clasificación o reconocimiento de patrones como pueden ser el reconocimiento de firmas o imágenes como rostros o dígitos y la categorización de textos. Una visión general de las SVM es la ofrecida por [Burges, 1998] o por [Cristianini y Shawe-Taylor, 2000], aunque existe una bibliografía muy extensa sobre este tema

2.1.1. Clasificador SVM lineal

Las SVM son clasificadores binarios lineales ya que se basan en encontrar un separador lineal o hiperplano que separe las dos clases basándose en una estrategia de maximización del margen. Esta técnica consiste en escoger entre todos los hiperplanos posibles a aquel que separe los patrones pertenecientes a dos clases distintas y cuya distancia a los vectores más próximos de cada clase sea máxima. Los vectores que determinan la frontera de cada clase se denominan vectores soporte.

2.1.1.1. Caso Linealmente Separable

En este tipo de problema nos enfrentamos a un conjunto de datos cuyos patrones pertenecen a dos tipos de clases que son fácilmente separables mediante un hiperplano lineal. Entre todos los hiperplanos separadores de estas dos clases etiquetadas como $y_i \in \{-1, +1\}$, existe un único hiperplano óptimo cuyo margen de separación es máximo. Este margen de separación es la distancia que hay desde el hiperplano separador a las muestras más cercanas que son denominadas vectores soporte.

Dicho hiperplano se define mediante la siguiente función:

$$f(x) = \underline{w}^T \cdot \underline{x} + b \tag{2.1}$$

Deseamos encontrar aquel hiperplano que iguale la función a cero de tal modo que podamos separar todos los puntos de acuerdo a la siguiente función, con la que se obtendrá una correcta clasificación para el caso de un conjunto separable:

$$\forall_i, \quad y_i(\underline{w}^T \cdot \underline{x}_i + b) > 0 \tag{2.2}$$

o también

$$\forall i, \qquad f(\underline{x}_i) = sign(\underline{w}^T \cdot \underline{x}_i + b) = \begin{cases} 1, & y_i = 1 \\ -1, & y_i = -1 \end{cases} \tag{2.3}$$

Según la expresión (2.3) podemos definir un hiperplano de margen para cada una de las clases, definiéndose para las muestras $\underline{x}$ denominadas vectores soporte: $\underline{w}^T \cdot \underline{x} + b = 1$ para la clase y_i=1 y $\underline{w}^T \cdot \underline{x} + b = -1$ para la clase y_i=-1.

A continuación se muestra un ejemplo de un conjunto de datos separable donde se puede ver el hiperplano óptimo y los hiperplanos de máximo margen que separan correctamente ambas clases:

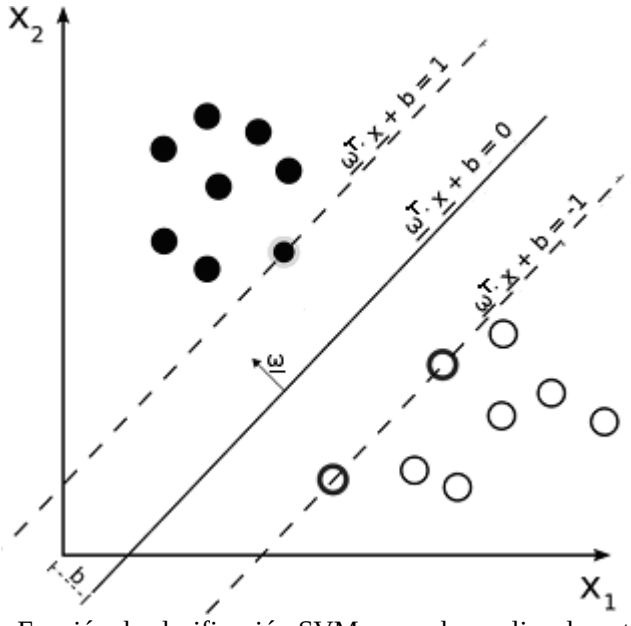


Figura 2.1: Función de clasificación SVM para el caso linealmente separable

No obstante, estas funciones (2.1), (2.2) y (2.3), son muy difíciles de optimizar computacionalmente ya que se debería tener en cuenta todos los valores posibles para *w* y *b* y finalmente quedarse con aquellos que maximicen el margen. Como esta opción resulta complicada, en la práctica se suele utilizar la siguiente función equivalente que es más sencilla de optimizar:

$$mín\ \frac{1}{2}\cdot\left\|\underline{w}\right\|^2 \tag{2.4}$$

Sujeto a las siguientes restricciones:

$$\forall_{i=0}^{n}:\ y_i(\underline{w}^T\cdot\underline{x}_i+b)\geq 1 \tag{2.5}$$

El uso de esta última función (2.4) sí resulta óptimo y hace que se reduzca el coste computacional de forma considerable. El principal inconveniente de esta función de optimización es que sólo puede utilizarse en problemas linealmente separables.

2.1.1.2. Caso No Linealmente Separable

En este tipo de problemas existe un conjunto cuyos datos que no pueden ser separados linealmente por lo que se define un clasificador de máximo margen blando o suavizado. En este caso se permite un cierto número de errores de clasificación. Por tanto, se incluye a las anteriores funciones de optimización un término de regularización que incluye el coste de equivocarse y un límite superior que indica el número de errores permitido, quedando la función de optimización del hiperplano:

$$mín\ \frac{1}{2}\cdot\left\|\underline{w}\right\|^2+C\cdot\sum_{i=1}^{n}\xi_i^d \tag{2.6}$$

Sujeto a las siguientes restricciones:

$$\begin{aligned}&\forall_{i=0}^{n}:\ y_i(\underline{w}^T\cdot\underline{x}_i+b)\geq 1-\xi_i\\&\forall_{i=0}^{n}:\ \xi_i>0\end{aligned} \tag{2.7}$$

donde:

C es el parámetro de regularización,
d toma un valor de 1 para un coste lineal y un valor de 2 para un coste cuadrático y
ξ_i son variables de holgura.

Las variables de holgura ξ_i son mayores de 1 si el ejemplo *i* se encuentra en el lado incorrecto del hiperplano separador y, por tanto, $\sum_{i=1}^{n}\xi_i$ define el límite superior de ejemplos que han sido mal clasificados.

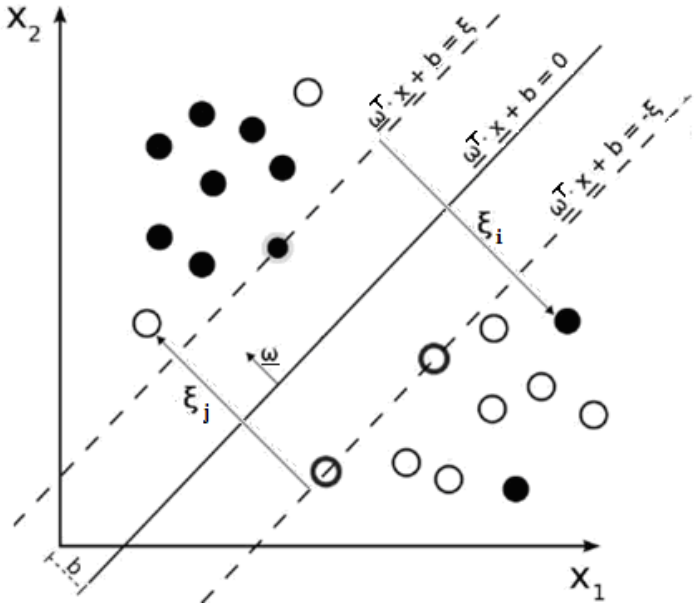


Figura 2.2: Función de clasificación SVM para el caso no linealmente separable

La variable de regularización *C* es un parámetro que permite una compensación entre en número de errores de clasificación y la complejidad del modelo. Un valor pequeño incrementa el número de errores permitido y un valor grande hace el comportamiento de este modelo se aproxime a la clasificación del caso linealmente separable o de margen duro.

2.1.2. Clasificador SVM no-lineal

Hasta el momento se han presentado reglas de clasificación lineal, pero una mayoría de los problemas que nos podemos encontrar en la realidad no pueden ser resueltos mediante clasificadores lineales ya que tienen una estructura no-lineal.

La clasificación por SVM puede ser transformada fácilmente a una clasificación no-lineal si se mapean los ejemplos del conjunto de datos a un espacio de características de mayor dimensionalidad mediante una transformación no lineal Φ.

2.1.2.1. Función kernel

En general, la transformación no-lineal de los datos no resulta computacionalmente muy eficiente, pero [Boser et al., 1992] descubrieron que la clasificación SVM tiene una propiedad especial que resuelve este problema. Esta propiedad de SVM hace que no sea necesario tener ningún conocimiento sobre la transformación Φ y, por tanto, este mapeo puede ser computado usando funciones *kernel*. Estas funciones satisfacen la condición del Teorema de Mercer y calculan el producto interno de los datos del espacio de entrada en el espacio de características del siguiente modo:

$$\textit{Función kernel:} \qquad K(\underline{x}_i, \underline{x}_j) = \Phi(\underline{x}_i) \cdot \Phi(\underline{x}_j) \tag{2.8}$$

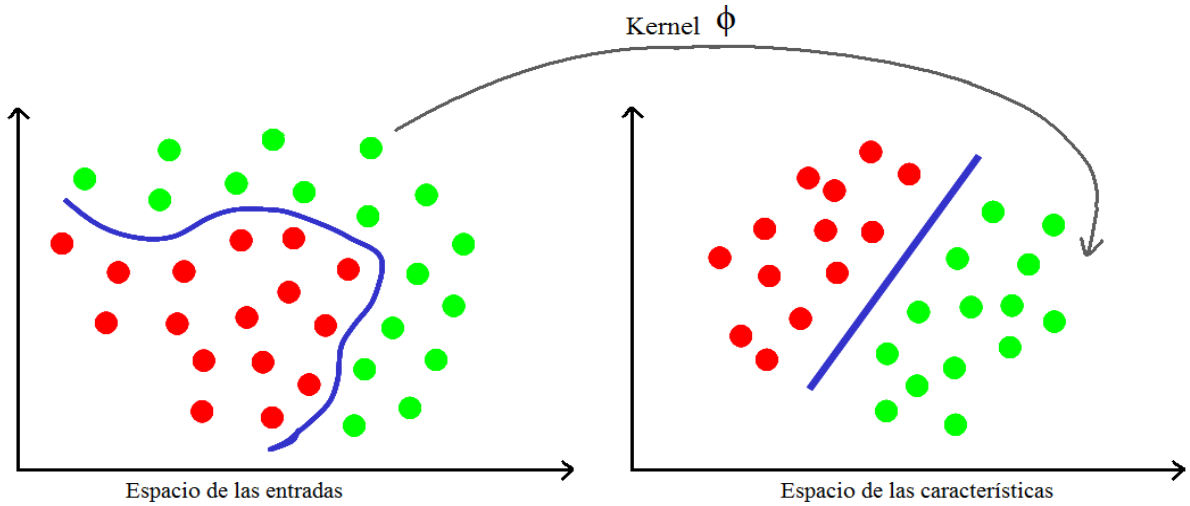


Figura 2.3: Transformación de un espacio de entrada linealmente no separable a un espacio linealmente separable mediante una función de kernel

Estas funciones de kernel se usan para construir clasificadores SVM para problemas en los que nos encontramos con un conjunto de datos con categorías no separables linealmente. Estas funciones redimensionan el espacio en el que se encuentran los datos haciendo que en el nuevo espacio al que lo trasladan sí que resulta linealmente separable. Es en el espacio transformado de características donde se halla el hiperplano óptimo de separación basado en las funciones de optimización por margen duro o blando. Más tarde, dicha redimensión se deshace transformando el hiperplano al espacio original y constituyendo la función de clasificación.

A continuación se muestran las funciones de kernel más utilizadas:

Lineal:

$$K(\underline{x}_i, \underline{x}_j) = \Phi(\underline{x}_i) \cdot \Phi(\underline{x}_j) = \underline{x}_i \cdot \underline{x}_j \tag{2.9}$$

Polinomial:

$$K(\underline{x}_i, \underline{x}_j) = (\underline{x}_i^T \cdot \underline{x}_j + 1)^d \tag{2.10}$$

Gaussiano o Función de Base Radial (RBF):

$$K(\underline{x}_i, \underline{x}_j) = \exp(-\gamma \cdot \left\| \underline{x}_i - \underline{x}_j \right\|^2) \tag{2.11}$$

Producto interior normalizado:

$$K(\underline{x}_i, \underline{x}_j) = \frac{\underline{x}_i \cdot \underline{x}_j}{|\underline{x}_i| \cdot |\underline{x}_j|} \tag{2.12}$$

Función sigmoide:

$$K(\underline{x}_i, \underline{x}_j) = \tanh(\gamma \cdot (\underline{x}_i^T \cdot \underline{x}_j) + \beta) \tag{2.13}$$

2.1.3. Ventajas de las SVM

Las máquinas de vector soporte tienen ciertas características que las han puesto en ventaja respecto a otras técnicas de clasificación y/o regresión.

La primera de ellas es que, al ser una técnica de aprendizaje automático, la máquina puede ir aprendiendo, a través de ejemplos, las salidas correctas para ciertas entradas.

En esta técnica no se requiere, aunque podría resultar interesante su uso, de una selección o reducción de términos. Esto es debido a que en el caso de que las muestras de una clase se distribuyan en zonas separadas del espacio, es decir, todos los puntos no están agrupados, es la transformación del espacio de entrada mediante funciones kernel la que solucione este problema.

Tampoco es necesario realizar un ajuste de parámetros en el caso de conjuntos linealmente separables ya que esta técnica posee su propio método para realizarlo.

Otras de sus fortalezas son que el entrenamiento es sencillo, no se consigue como solución un óptimo local, situación que se logra en otras técnicas de clasificación como son las redes neuronales. También, se pueden escalar relativamente bien para datos en espacios de alta dimensionalidad y puede ser controlado explícitamente el compromiso entre el error de clasificación y la complejidad del método mediante algunos términos de regularización.

2.1.4. Desventajas de las SVM

La principal desventaja de estos métodos es su tendencia al sobreentrenamiento que ocurre cuando se han aprendido muy bien los datos durante la fase de entrenamiento pero en la fase de test no se clasifican bien ejemplos nunca antes vistos. Por lo tanto, no se consigue una buena generalización del modelo.

Otros inconvenientes de estas técnicas son que, por un lado, la predicción del clasificador no tiene ningún significado probabilístico y, por otro lado, se necesita elegir bien la función de kernel que se va a utilizar ya que éstas deben satisfacer las condiciones del Teorema de Mercer.

2.2. Clasificación con Máquinas de Vectores Soporte Multiclase

Dado que SVM sólo resuelve problemas de naturaleza binaria surge la necesidad de implementar algunas técnicas para solucionar problemas de clasificación con múltiples clases ya que éstos aparecen comúnmente en los dominios de aplicación en los que nos encontramos.

2.2.1. Aproximación Directa

Con este objetivo, se han propuesto diversas aproximaciones en este sentido. Como aproximación directa fueron [Weston y Watkins, 1999] quienes propusieron una modificación de las máquinas de vectores soporte para tareas multiclase.

Se define un clasificador SVM multiclase para *k* categorías en cuya fase de entrenamiento calcula varios hiperplanos y con cada uno de ellos se separa los datos correspondientes a esa clase del resto. En la fase de test, al realizar predicciones para cada nuevo patrón, el clasificador es capaz de establecer un margen sobre cada uno de los hiperplanos. Estos márgenes hacen referencia a la confianza que se tiene sobre si un ejemplo pertenece a cada una de las clases. Como decisión final, el clasificador ofrece como predicción aquella clase que maximiza el margen.

Por tanto, se modifica la función de optimización del problema SVM clásico para el caso linealmente no separable (2.6) para tener en cuenta las *k* clases:

$$mín\ \frac{1}{2}\cdot\sum_{m=1}^{k}\left\|\underline{w}_m\right\|^2 + C\cdot\sum_{i=1}^{N}\sum_{m=1}^{k}\xi_i^m \tag{2.14}$$

Sujeto a:

$$\begin{aligned}&\forall_{i=0}^{n},\ \forall m \neq y_i :\ \underline{w}^T{}_{y_i}\cdot\underline{x}_i + b_{y_i} \geq \underline{w}^T{}_m\cdot\underline{x}_i + b_m + 2 - \xi_i^m\\ &\forall_{i=0}^{n},\ \forall_{m=1}^{k} :\ \xi_i^m \geq 0\end{aligned} \tag{2.15}$$

2.2.2. División del problema multiclase en subproblemas binarios

Otra técnica muy utilizada y de fácil aplicación para conseguir una aproximación a SVM multiclase, es dividir el problema con múltiples clases y convertirlo en varios problemas binarios. En este sentido, existen varios métodos que permiten la combinación de los distintos clasificadores SVM binarios siendo los más comunes el 1-contra-todos (*1-vs-r* o *1-vs-All*), el 1-contra-1 (*1-vs-1*) o los grafos dirigidos DAG.

2.2.2.1. Caso 1-contra-todos (1-vs-All)

En esta técnica se construyen tantos clasificadores binarios como clases haya en el problema, *N*. Cada clasificador define un hiperplano que separa las muestras de la clase *i* de las muestras de las *N-1* clases restantes.

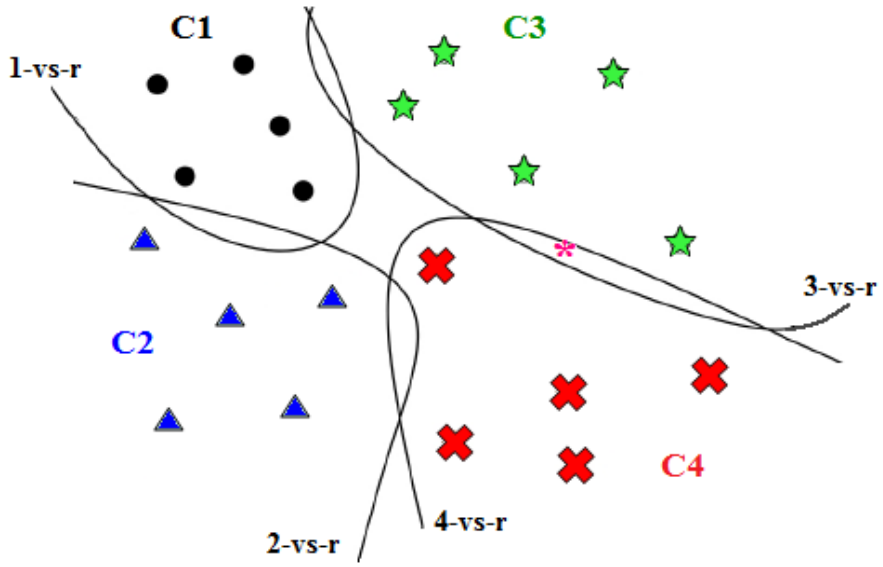


Figura 2.4: Ejemplo de fronteras para clasificación *1-vs-All* para un problema con 4 clases. El patrón * es el nuevo patrón que va a ser clasificado

Cada patrón de prueba es clasificado por cada uno de los clasificadores *i-vs-r*, que predicen y asignan a dicho patrón una determinada categoría, C_i o $\overline{C_i}$. La decisión final, tras la combinación de todos los clasificadores *1-vs-All*, es aquella con la que se proporcione el máximo margen aplicando la siguiente ecuación:

$$\hat{C}_i = \arg \max_{m=1,\ldots,k} (\underline{w}^T{}_m \cdot \underline{x}_i + b_i) \tag{2.16}$$

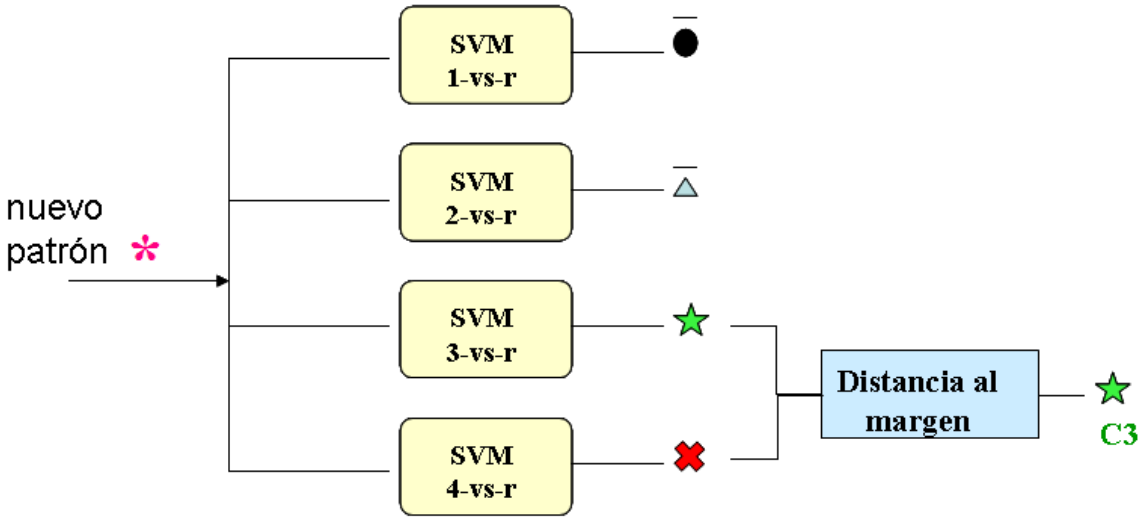


Figura 2.5: Ejemplo de clasificación para un problema de 4 clases combinando 4 clasificadores *1-vs-All*. El nuevo patrón * es clasificado como de clase 3 por tener mayor distancia al margen**.**

2.2.2.2. Caso 1-contra-1 (Pairwise)

En esta técnica se construyen tantos clasificadores como parejas de clases haya en el problema, $\frac{N(N-1)}{2}$ siendo N el número de clases. De esta manera se consigue enfrentar todas las clases una a una en donde cada hiperplano separa las muestras de la clase *i* de las muestras de la clase *j*.

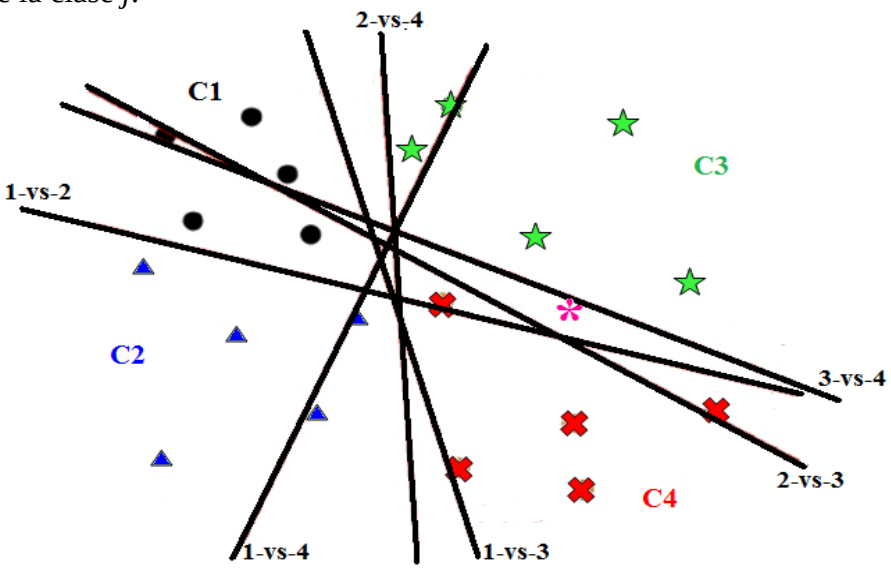


Figura 2.6: Ejemplo de fronteras para clasificación *pairwise* para un problema con 4 clases. El patrón * es el nuevo patrón que va a ser clasificado

Cada patrón de prueba es clasificado por cada uno de los clasificadores pareados *i-vs-j* que le asignan una determinada categoría, C_i o C_j. La decisión final, tras la combinación de todos los clasificadores, se hace mediante un sistema de votación por lo que la categoría asignada es aquella que haya sido más veces votada o asignada por los clasificadores pareados.

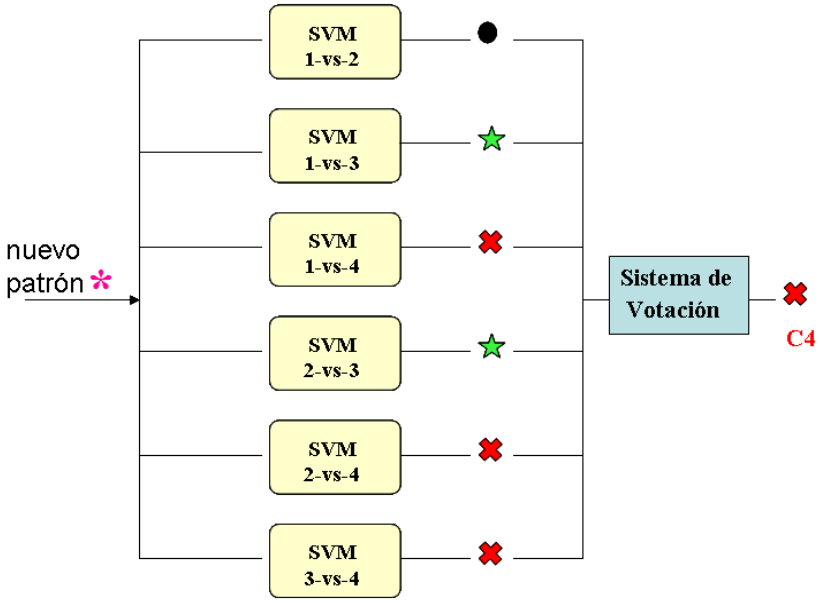


Figura 2.7: Ejemplo de clasificación para un problema de 4 clases combinando 6 clasificadores pareados El nuevo patrón * es clasificado como de clase 4 por mayoría de votos.

2.2.2.3. Caso por Grafos Dirigidos (DAG)

Otra de las técnicas de aproximación a SVM multiclase es la basada en grafos DAG o *Directly Acyclic Graphs*. Es una derivación de la técnica *Pairwise* (*1-vs-1*) en la que se modifica el proceso final de decisión durante la fase de clasificación.

De esta manera, como en el caso *pairwise*, se construyen tantos clasificadores como parejas de clases haya en el problema, $\frac{N(N-1)}{2}$, y se crean con ellos un grafo en forma de árbol. En el proceso de decisión, cada patrón va recorriendo todos los nodos de dicho árbol donde se va decidiendo progresivamente a que clase pertenece. Finalmente se asigna la clase que decide el clasificador "hoja".

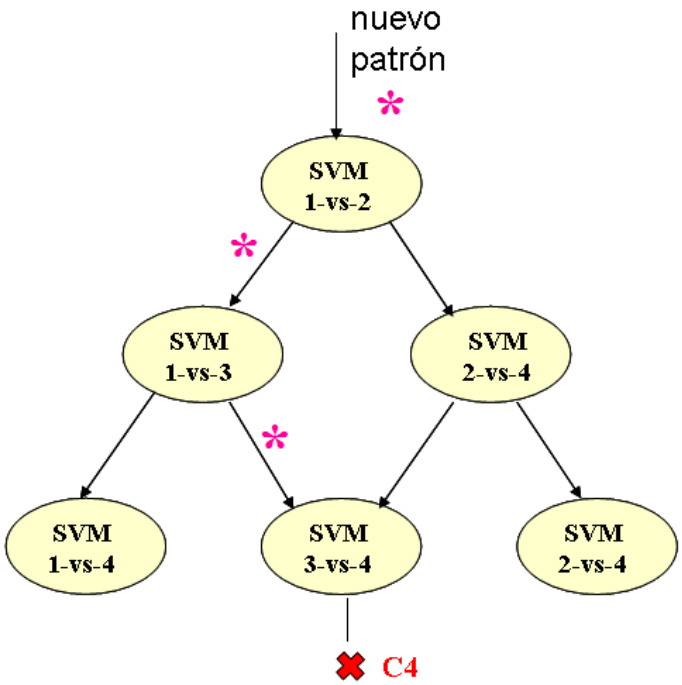


Figura 2.8: Ejemplo de clasificación *DAG* para un problema de 4 clases combinando 6 clasificadores pareados *1-vs-1* en forma de árbol. El nuevo patrón * recorre todos los nodos del árbol y es clasificado como de clase 4 por el nodo hoja.

2.2.2.4. Comparación de estas Técnicas de Multiclasificación

A la hora de comparar las tres técnicas de combinación mencionadas anteriormente en el caso *1-vs-All* hay que entrenar un menor número de clasificadores binarios con respecto a los otros dos casos, N y $\frac{N(N-1)}{2}$ respectivamente, siendo N el número de clases del problema. No obstante, en los casos *1-vs-1* y *DAG*, aunque hay que entrenar un mayor número de clasificadores, éstos hay que entrenarlos con un número más reducido de muestras. También en estos casos los clasificadores binarios poseen una información más precisa, ya que se tiene en cuenta exactamente a que clase pertenece realmente cada muestra.

Hay que tener en cuenta que la técnica *DAG* tiene un principal inconveniente con respecto a las otras dos técnicas y es que es muy sensible al clasificador pareado que se escoja para ser el nodo raíz del árbol. Esto es debido a que si el clasificador raíz no es muy bueno, el clasificador final se equivoca en muchas ocasiones en la clasificación de los patrones y error aumenta notablemente. Por esa razón, hay que elegir correctamente a los clasificadores del nodo raíz y de las capas superiores escogiendo a aquellos que cometan menos errores de clasificación en el entrenamiento o a los que tengan los mayores márgenes.

Numerosos estudios, como los realizados por [Hsu y Lin, 2002], dicen que la técnica *Pairwise* es la que da mejores resultados en la práctica aunque también existen otras investigaciones, como las publicadas por [Rifkin y Klautau, 2004], que defienden la utilización de la técnica *1-vs-All* en problemas multiclase.

En este último estudio, se comparan ambos esquemas de clasificación en base del tiempo de entrenamiento y de test que emplean. Con respecto al tiempo de entrenamiento, en el caso de que se tengan conjuntos de datos muy grandes, el esquema *1-vs-All* es mucho más lento que el esquema *1-vs-1*, aunque estos autores demuestran que el tamaño del conjunto no marca de manera definitiva las diferencias entre ambos esquemas. Con respecto al tiempo de test, en muchas aplicaciones el tiempo empleado para realizar las pruebas es mucho más importante que el empleado para entrenar el sistema. En este caso el tiempo de test para los esquemas *1-vs-All* suelen ser peores que en los esquemas *1-vs-1*, sobre todo cuando los conjuntos son de gran tamaño, pero las diferencias no suelen ser muy grandes.

Para finalizar, cabe pensar que dependiendo de los recursos que posea el usuario o las necesidades y prestaciones que se desea conseguir, el uso de una u otra técnica de clasificación pueda resultar más o menos beneficioso.

2.3. Presentación del Artículo

En nuestro estudio, nos hemos basado en el artículo *"Reducing the number of sub-classifiers for pairwise multi-category support vector machines"* de [Ye y Shang-Teng, 2007]. En esta investigación los autores proponen un nuevo método para problemas de clasificación multiclase basado en clasificadores SVM.

Con este nuevo método intentan resolver los inconvenientes que aparecen en los otros algoritmos de multiclasificación, en *1-vs-todos* un gran coste temporal en el entrenamiento y regiones inclasificables, en *pairwise* y *DAGSVM* la necesidad de entrenar numerosos clasificadores.

2.3.1. Estrategia utilizada: Muestreo basado en incertidumbre

En el algoritmo que proponen los autores utiliza una estrategia basada muestreo de incertidumbre. Esta estrategia fue propuesta por [Lewis y Gale, 1994] y es utilizada, dentro

de las teorías de aprendizaje activo, para seleccionar sólo las muestras "útiles" mediante la medida de su incertidumbre hacia el clasificador actual

La principal idea de esta estrategia es que el clasificador se beneficiará de las muestras entrenadas que son más inciertas para el clasificador actual. Para llevarla a cabo, se requiere de un clasificador probabilístico que asigne a las muestras no etiquetadas una etiqueta con una cierta probabilidad. Las muestras no etiquetadas que son más inciertas son seleccionadas por el clasificador para ser predichas y etiquetadas.

2.3.2. Método US-MSVM (*Uncertainty sampling-based multi-SVM*)

Los autores utilizan un muestreo de incertidumbre para realizar una clasificación multiclase basada en SVM. Se aplica dicha estrategia para seleccionar aquellos patrones que permitan decidir cuales son las categorías que son más indistinguibles para así entrenar únicamente los clasificadores que sean más "útiles" ignorando los que son triviales. Por tanto, solamente algunos clasificadores son seleccionados para ser entrenados y no es necesario entrenar tantos clasificadores SVM como en el caso *pairwise*, $\frac{N(N-1)}{2}$, siendo N el número de clases del problema.

2.3.2.1. Fase de entrenamiento

En la fase de entrenamiento, gradualmente, se escoge el clasificador más útil de acuerdo a dicha estrategia. Esta fase es una fase iterativa en la que se entrenan una serie subclasificadores hasta que se cumplan unas determinadas restricciones fijadas previamente: el número máximo de clasificadores a entrenar y una distancia de umbral.

Primero, se elige un par de clases de manera aleatoria y con los patrones pertenecientes a éstas se entrena un subclasificador SVM. Éste decide la etiqueta de los datos, +1 ó -1, y se calcula la probabilidad de muestras positivas o PPS para los ejemplos de cada una de las clases del subconjunto de prueba del problema utilizando la siguiente expresión:

$$PPS_{k,i} = \frac{card\left(f_k(x^i) > 0 \right)}{card\left(x^i\right)} \tag{2.17}$$

siendo $\left\{\underset{\sim}{x}^i\right\}_i$ el conjunto de muestras de la categoría *i*, la función *card(·)* indica el número de elementos que cumple una determinada condición y $f_k(\cdot)$ la función de decisión del subclasificador SVM *k*.

Para realizar la estrategia de muestreo de incertidumbre, se define la matriz de decisión DM cuya columna *i* es el vector de PPS para la clase *i* para cada uno de los *k* subclasificadores SVM.

$$DM = \begin{bmatrix} PPS_{1,1} & PPS_{1,2} & \cdots & PPS_{1,i} & \cdots & PPS_{1,N} \\ PPS_{2,1} & PPS_{2,2} & \cdots & PPS_{2,i} & \cdots & PPS_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ PPS_{k,1} & PPS_{k,2} & \cdots & PPS_{k,i} & \cdots & PPS_{k,N} \end{bmatrix} \quad (2.18)$$

Con esta matriz se puede medir la incertidumbre entre todas las categorías ya que si dos columnas se parecen, las dos clases a las que pertenecen son indistinguibles para todas los subclasificadores entrenados. La medida de similitud entre dos columnas se realiza con la distancia Euclídea. Las clases más indistinguibles, las de menor distancia, se eligen para ser el próximo par de entrenamiento en la siguiente iteración. Las iteraciones continúan hasta que la mínima distancia entre todos los pares de categorías supere la distancia umbral o se supere el máximo número de clasificadores entrenados.

Paso 1: Especificar el máximo número de clasificadores que se podrán entrenar, r, y la distancia umbral que no podrá traspasarse, d^*.
Establecer k=1; Seleccionar 2 categorías C_{j1} y C_{j2} aleatoriamente; Establecer el par de entrenamiento $Trained\_Pair = \left\{ \left\langle C_{j1}, C_{j2} \right\rangle \right\}$

Paso 2: Seleccionar las muestras cuyas categoría es C_{j1} o C_{j2} para hacer un nuevo conjunto de entrenamiento; Entrenar un subclasificador SVM l_k con dicho subconjunto.

Paso 3: Predecir un subconjunto de muestras usando el subclasificador l_k , calcular el valor $PPS_{k,i}$ para cada categoría, y añadir el vector PPS_k como una nueva fila de la matriz de decisión DM.

Paso 4: Medir las distancias de todos los pares de categorías utilizando las columnas de la matriz DM, exceptuando los pares de clases que ya han sido añadidos a Trained_Pair, usando la distancia euclídea según la siguiente expresión:

$$dist_{i,j} = \left\| \begin{bmatrix} PPS_{1,i} \\ PPS_{2,i} \\ \vdots \\ PPS_{k,i} \end{bmatrix} - \begin{bmatrix} PPS_{1,j} \\ PPS_{2,j} \\ \vdots \\ PPS_{k,j} \end{bmatrix} \right\|$$

Seleccionar las 2 clases que tienen mayor incertidumbre como las nuevas C_{j1} y C_{j2}, y añadir el nuevo par $\left\langle C_{j1}, C_{j2} \right\rangle$ *a Trained_Pair. Incrementar k.*

Paso 5: Repetir el proceso desde los Pasos 1 al 4 hasta que se alcanza el número máximo de clasificadores, k=r, o la mínima distancia de todos los pares de categorías es mayor que la distancia umbral d^*.

Figura 2.9: Algoritmo de Entrenamiento del método US-MSVM

2.3.2.2. Fase de Prueba

En la fase de prueba o test, cada subclasificador predice si la etiqueta del patrón de test es +1 ó 0. Este vector de decisiones se compara con cada una de las columnas de la matriz de decisión y si ambos son similares, la columna de dicha categoría y el vector predicho son los más indistinguibles para todos los clasificadores entrenados. Por esta razón, se asigna aquella clase con la que se obtenga la menor distancia ya que esta será la categoría correcta con una mayor probabilidad.

Paso 1: Cada subclasificador k predice si la muestra de test es positiva o negativa. Se crea el vector PY cuyos elementos son la predicción de cada subclasificador, del siguiente modo:

$$PY = [py_1 \quad py_2 \quad \cdots \quad py_R], \qquad siendo \quad py_i = \begin{cases} 1, & f(x) \geq 0 \\ 0, & f(x) < 0, \end{cases}$$

siendo R el número de clasificadores que han sido entrenado, $R \leq r$.

Paso 2: Se mide las distancias de cada columna de la matriz DM y el vector PY^T. Se predice la categoría que tengan mayor incertidumbre utilizando la siguiente expresión:

$$categoria = \arg\min_i \left\| PY^T - [PPS_{1,i} \quad PPS_{2,i} \quad \cdots \quad PPS_{R,i}]^T \right\|$$

Figura 2.10: Algoritmo de Test del método US-MSVM

2.3.3. Resultados Experimentales

Los autores realizan experimentos de su nuevo método US-MSVM, sobre tres conjuntos de datos “Reuters-21578”, “20-NG” y “USPS” y comparan los resultados que obtienen con los de pairwise. Como parámetros del clasificador SVM para ambos métodos se eligen el coste C=100 y un kernel tipo RBF con γ=0.1 y como restricciones del experimento para US-MSVM se eligen la distancia umbral d^*=0.8 y el número máximo de clasificadores a entrenar $r = 0.5 * \frac{N(N-1)}{2}$.

Como medidas de evaluación y comparación entre los distintos métodos utilizan las medidas de Recuperación de la Información: Precisión, Recuperación y medida F_1. A continuación se muestran las medidas del F_1 para los métodos US-MSVM y *Pairwise*, en los tres conjuntos del experimento (Tablas 5.3, 5.4 y 5.5 del artículo):

	Earn	acq	Money-fx	Grain	Crude	Trade	Interest	Ship
US-MSVM	98,0%	93,7%	70,9%	95,4%	83,1%	78,8%	72,9%	76,1%
Pairwise	98,3%	95,7%	74,5%	94,6%	82,9%	79,3%	73,2%	76,5%

Tabla 2.1: Comparación de la medida F_1 para Reuters con r=14 (Tabla 5.3 del artículo.)

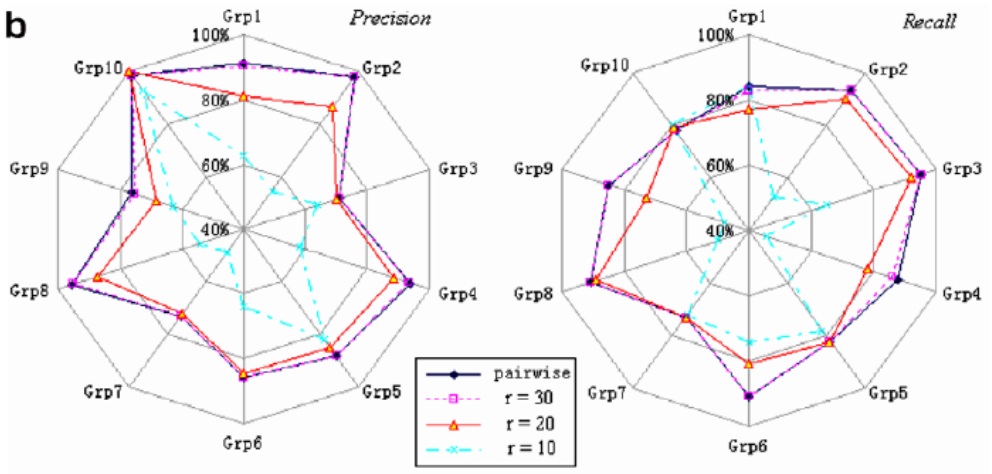
Grupo	Grp 1	Grp 2	Grp 3	Grp 4	Grp 5	Grp 6	Grp 7	Grp 8	Grp 9	Grp 10
US-MSVM	87,5%	93,8%	81,2%	86,9%	84,5%	86,1%	73,0%	91,4%	78,6%	87,2%
Pairwise	87,5%	95,4%	81,2%	90,7%	85,0%	88,5%	73,0%	93,3%	80,2%	87,2%

Tabla 2.2: Comparación de la medida F1 para “20NG” con r=23 (Tabla 5.4 del artículo.)

Grupo	“0”	“1”	“2”	“3”	“4”	“5”	“6”	“7”	“8”	“9”
US-MSVM	93,5%	95,8%	88,0%	89,1%	93,1%	86,9%	95,5%	91,7%	90,3%	94,2%
Pairwise	93,5%	96,2%	88,6%	90,0%	95,6%	89,3%	96,0%	92,4%	91,5%	95,5%

Tabla 2.3: Comparación de la medida F1 para “USPS” con r=23 (Tabla 5.5 del artículo.)

En los experimentos se estudia la influencia del número máximo de clasificadores que hay que entrenar en los resultados en comparación con los obtenidos mediante *Pairwise*. Concluyen que para valores pequeños, las medidas para US-MSVM son pobres, mientras que cuando se incrementa este número, sobretodo en valores por encima de la mitad de clases del problema, dichas medidas mejoran y se parecen más a las conseguidas con pairwise. Esto es debido a que en US-MSVM los clasificadores son entrenados en orden de importancia siendo los más “útiles” los primeros en entrenarse, por lo que, cuando el número de clasificadores es mayor que un cierto umbral (la mitad de las clases totales) los clasificadores entrenados a partir de ese momento se consideran casi “inútiles”. En la Figura 2.9 se muestra la Fig.3 del artículo donde puede comprobarse la influencia del número de clasificadores en los resultados de precisión y exhaustividad.



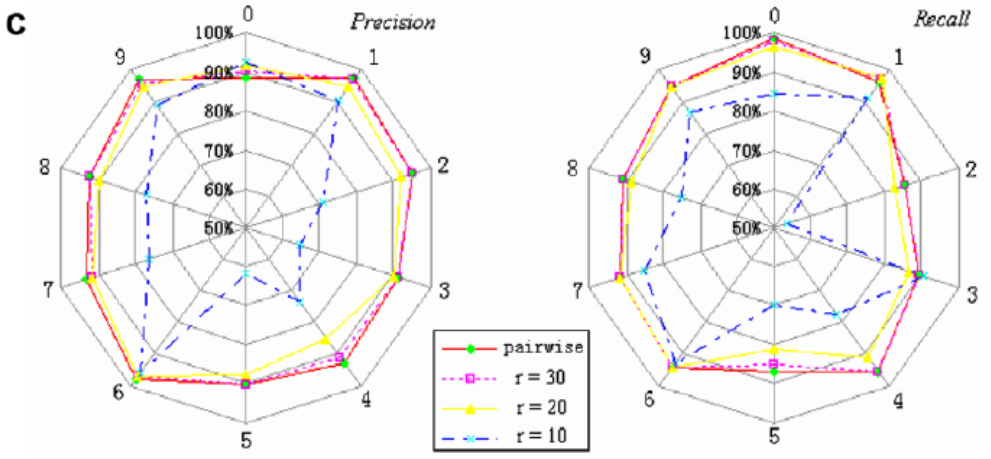


Figura 2.11: Fig.3 del artículo que muestra la influencia del número de clasificadores en los resultados de *precision* y *recall* para los conjuntos *10Newsgroups* (b) y *USPS* (c)

Por último, los autores concluyen, basándose en los resultados experimentales del artículo, que para los conjuntos de datos de palabras reales el método US-MSVM puede sustituir a la técnica *Pairwise* ya que se obtienen resultados comparables. Y el nuevo método tiene como ventaja que su fase de entrenamiento es mucho menor que en *Pairwise* ya que se reduce el número de clasificadores que se deben entrenar, reduciéndose por tanto el coste computacional.

Capítulo 3

Descripción de los Métodos Propuestos.

Debido al coste computacional y temporal de las técnicas clásicas de clasificación en problemas multiclase, en este capítulo se van a presentar dos aproximaciones basadas en clasificación por máquinas de soporte vectorial para intentar reducir dicha carga. La primera aproximación será deconstructiva basada en poda de clasificadores SVM pareados y la segunda será una aproximación constructiva.

Para comenzar, haremos mención a diferentes técnicas de combinación de clasificadores que se utilizarán en este proyecto para predecir la categoría a la que pertenece el conjunto de patrones de prueba.

Posteriormente, se explicarán las diferentes técnicas de deconstrucción y eliminación de clasificadores que se van a analizar en este proyecto y, finalmente, se expondrán varias estrategias de construcción basadas en las anteriores.

Con estas técnicas se van a realizar diversos experimentos con el fin de reducir la complejidad de los métodos de multiclasificación tradicionales y cuyos resultados se analizarán en capítulos posteriores.

3.1. Estrategias de combinación o fusión de clasificadores

La combinación de varios clasificadores se utiliza, principalmente, para mejorar los resultados y prestaciones que se consigue con cada uno de los clasificadores individuales.

En un principio, la fusión de clasificadores asume que todos los clasificadores individuales son competitivos, es decir, en la predicción se van a considerar igualmente expertos. Bajo ese aspecto, cada clasificador emite una decisión para cada uno de los patrones del conjunto de prueba. Dichos vectores de decisiones pueden contener a su vez diferentes informaciones:

- Información acerca de las estimaciones de las probabilidades a posteriori con las que el clasificador individual decide a que clase pertenece cada patrón.

- Información acerca de la etiqueta que cada uno de los clasificadores individuales asigna a cada patrón.

- Información, únicamente, sobre si la decisión de cada clasificadores es correcta o incorrecta. Este tipo de vector es conocido como "oráculo", ya que para saber si la decisión tomada por el clasificador ha sido correcta o no, se debe tener previamente conocimiento acerca de la etiqueta real de cada patrón.

En nuestros experimentos se van a utilizar clasificadores SVM obtenidos de la librería *LIBSVM*[a]. Un clasificador binario formado por dos clases etiquetadas como $y_i \in \{-1, 1\}$ ofrece como salida información para cada patrón del conjunto de prueba sobre las etiquetas que han sido predichas y asignadas.

También, en este clasificador SVM binario se ofrece información sobre las probabilidades a posteriori de que un determinado patrón x_i pertenezca a cada clase:

$$Prob\,(\, y_i = +1 \,|\, x_i \,) \tag{3.1}$$

En una nota de los autores de esta librería [Lin et al., 2007] se comenta que las probabilidades a posteriori son calculadas mediante una aproximación utilizando la siguiente función sigmoide:

$$Prob\,(\, y_i = +1 \,|\, x_i \,) \approx P_{A,B}(f) \equiv \frac{1}{1+\exp(Af + B)} \tag{3.2}$$

siendo $f = f(x)$ la función de decisión del clasificador binario SVM por la que se predice la etiqueta o clase de cualquier ejemplo de prueba. Esta función puede verse en la sección 2.1 de clasificación de máquinas de vector soporte

Dado el conjunto de todas las salidas proporcionadas por cada uno de los clasificadores individuales se debe tomar una decisión para el clasificador final o combinado. Para ello, existen varias estrategias de combinación o fusión de clasificadores. A continuación se mencionarán algunas de las más comúnmente utilizadas.

3.1.1. Voto por mayoría simple o *Maxwins*

En primer lugar, esta técnica es una de la más utilizadas en la fusión de clasificadores. Cada uno de los clasificadores individuales proporciona una decisión con la etiqueta de la clase que ha sido asignada a cada patrón. La decisión final de la clase asignada se obtiene como la más "votada", es decir, aquella que es decidida por la mayoría de los clasificadores.

[a] - Se puede descargar la librería en: http://www.csie.ntu.edu.tw/~cjlin/libsvm

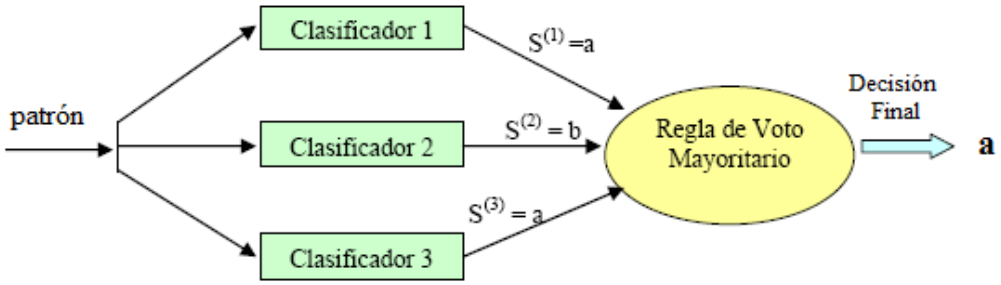


Figura 3.1: Ejemplo de decisión utilizando la combinación de 3 clasificadores por voto por mayoría. Al patrón de muestra se le asigna la clase a por ser la más votada

Cuando se trabaja con conjuntos de datos con más de dos clases, usualmente ocurren empates entre algunas de ellas. Para resolver este problema, se han considerado varios criterios:

- Seleccionar de forma aleatoria a la clase ganadora, de entre las clases más votadas.
- Seleccionarla con la implementación de un clasificador adicional cuya función es la de inclinar la decisión hacia una determinada clase.
- Escoger aquella clase que haya sido asignada con mayor probabilidad a posteriori por los clasificadores individuales.

En nuestros experimentos se va a utilizar esta técnica de votación por mayoría utilizando la información de las decisiones o etiquetas que ofrece cada clasificador SVM pareado. Para cada patrón cada clasificador pareado decide la etiqueta de la clase a la que pertenece por lo que esa clase recibirá un voto y al final se le asigna la clase más votada. En caso de producirse algún empate de elige aquella clase que haya sido asignada con mayor probabilidad media por los clasificadores.

3.1.2. Voto por mayoría ponderada

Una mejora del método de voto simple es el *voto por mayoría ponderada*. Dicha mejora consiste en que a cada clasificador individual se le asigna un determinado peso que pondera la confianza de su decisión en relación al resultado que proporcionan, por lo general, se asigna de acuerdo a la estimación de la probabilidad de error o por su precisión o eficacia.

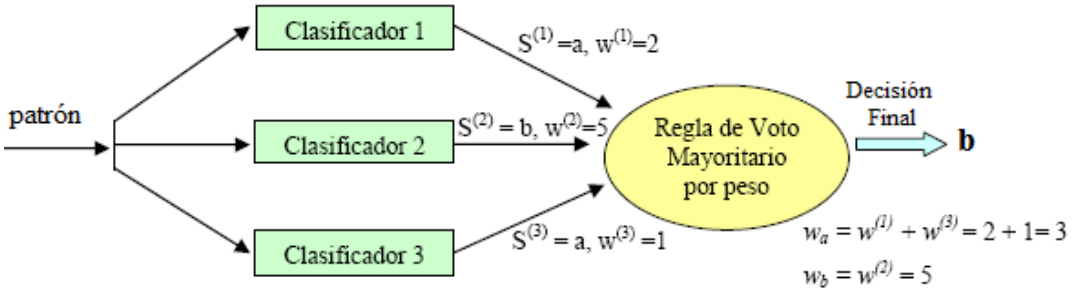


Figura 3.2: Ejemplo de decisión utilizando la combinación de 3 clasificadores por voto por mayoría ponderada. Al patrón de muestra se le asigna la clase b que aunque no es la más votada si es con la que mayor peso se ha decidido

3.1.2.1. MaxWins Votos

En nuestros experimentos vamos a utilizar un método de voto por mayoría ponderada al que hemos denominado *MaxWins Votos* y que puede resultar útil en el caso de que no se tengan el mismo número de clasificadores, y por tanto distinto número de votos, para cada una de las clases que existen en el problema. En este método se asigna un determinado peso a cada uno de los clasificadores individuales teniendo en cuenta para la combinación final siempre el máximo número de clasificadores o votos posible para cada clase.

En este sentido, en un caso en el que no se tienen el mismo número de clasificadores para cada clase si utilizamos para la combinación el método de voto por mayoría simple, la clases que tenga menos clasificadores recibirá menor número votos por lo que aumentará su error de clasificación y, de este modo, se reducirá la eficiencia de la decisión final del clasificador combinado. Por esta razón, si aplicamos un peso a cada clase teniendo en cuenta siempre el número máximo de clasificadores o votos posibles para las mismas, la eficiencia del método aumentará.

Por ejemplo, en un problema de 4 clases se tienen 6 clasificadores binarios. Para decidir la clase para un determinado patrón de clase real C_1 vamos a realizar la decisión final utilizando las técnicas de voto mayoritario simple y ponderado mencionadas antes.

Para un determinado patrón de prueba, cada clasificador ha decidido a que clase pertenece según la siguiente configuración:

Clasificadores binarios	**C_1-C_2**	**C_1-C_3**	**C_1-C_4**	**C_2-C_3**	**C_2-C_4**	**C_3-C_4**
Clase decidida por cada clasificador	1	1	1	2	2	3

Tabla 3.1: Ejemplo de clases decididas por cada uno de los 6 clasificadores pareados para un problema de 4 clases

Entonces, tenemos los siguientes valores del número de votos, número de votos posible y el peso para cada clase:

Clase	**C1**	**C2**	**C3**	**C4**
Numero de Votos por Clase	3	2	1	0
Número de Votos Posibles por Clase	3	3	3	3
Peso por Clase (nº Votos recibidos por clase/nº votos posibles por clase)	3/3=1	2/3	1/3	0/3=0

Tabla 3.2: Ejemplo de votos, votos posibles y peso asignado para cada una de las 4 clases

Por tanto, si se utiliza para designar la clase del patrón de prueba la técnica de voto mayoritario simple se decide que el patrón es de clase C_1 dado que es la clase que obtiene el mayor número de votos. En el caso de utilizar la técnica de voto por mayoría ponderada se asignaría también la clase C_1 pero esta vez dado que el peso para esta clase es el mayor. En este supuesto, en ambos casos se decide que el patrón pertenece a la clase C_1, por lo que no se comete error.

Por el contrario, imaginemos que se ha eliminado un clasificador, por ejemplo, el C_1-C_3, por lo que no se están utilizando el mismo número de clasificadores para cada clase.

Entonces en este momento, para el mismo patrón de prueba, tendríamos las siguientes decisiones para cada clasificador:

Clasificadores binarios	C_1-C_2	C_1-C_4	C_2-C_3	C_2-C_4	C_3-C_4
Clase decidida por cada clasificador	1	1	2	2	3

Tabla 3.3: Ejemplo de clases decididas para un problema de 4 clases para cada uno de los 5 clasificadores pareados después de haber eliminado el clasificador C_1-C_3

Y tendríamos tenemos los siguientes valores del número de votos, número de votos posible y el peso para cada clase:

Clase	C1	C2	C3	C4
Numero de Votos por Clase	2	2	1	0
Número de Votos Posibles por Clase	2	3	2	3
Peso por Clase (nº Votos recibidos por clase/nº votos posibles por clase)	2/2=1	2/3	1/2	0/3=0

Tabla 3.4: Ejemplo de votos, votos posibles y peso asignado para cada una de las 4 clases tras haber eliminado el clasificador C_1-C_3

En este caso, en el método de voto por mayoría simple se produce un empate ya que tanto la clase C_1 como la clase C_2 han recibido dos votos cada una. Esto ha sucedido ya que ha reducido el número de votos posible para la clase C_1 (la real del patrón). En cambio utilizando el método de votos por mayoría ponderada se decide la clase C_1 ya que el peso para esta clase sigue siendo el mayor. En este caso no afecta que se haya reducido el número de clasificadores para la clase C_1 ya que al no tener en cuenta solamente el voto para cada clase sino su ponderación con el número de votos posibles se sigue asignando el valor correcto de la categoría para el patrón de prueba.

3.1.3. Métodos de nivel de confianza

En último lugar, vamos a estudiar las estrategias de fusión o combinación de clasificadores basadas en la medida de algunos valores de confianza que ofrecen como salida dichos clasificadores.

Algunos clasificadores, como ya hemos visto, pueden ofrecer como salida de cada uno de los patrones de entrada información acerca de valores de confianza o medidas de distancias. Estas medidas pueden ser interpretadas como la probabilidad de que un patrón pertenezca a las diferentes categorías existentes en el problema.

Estas técnicas de combinación se basan en calcular una nueva medida de confianza utilizando reglas fijas y operadores estadísticos para la decisión final. Se puede utilizar como medida la estimación de las probabilidades a posteriori con las que los clasificadores individuales deciden a qué clase pertenece cada patrón.

Siendo C el número de clases, N el número de clasificadores y f_i^j el valor de la probabilidad a posteriori del clasificador i para el patrón j, se pueden aplicar las siguientes reglas:

3.1.3.1. Máximo

Para cada patrón se calcula el máximo valor de las probabilidades de una misma clase que haya sido asignada en todos los clasificadores individuales y se asigna la clase con el máximo valor.

$$\forall patron\ j, \qquad y_j = \arg\max_{i=1,\ldots,C} \left(\max_{\forall n\ asignada} \left(f_1^j, \ldots, f_N^j \right) \right) \tag{3.3}$$

3.1.3.2. Mediana

Para cada patrón se calcula el valor que es la mediana de las probabilidades de una misma clase que haya sido asignada en todos los clasificadores individuales y se asigna la clase con el máximo valor.

$$\forall patron\ j, \qquad y_j = \arg\max_{i=1,\ldots,C} \left(med_{\forall n\ asignada} \left(f_1^j, \ldots, f_N^j \right) \right) \tag{3.4}$$

3.1.3.3. Suma

Para cada patrón se calcula la suma de las probabilidades de una misma clase que haya sido asignada en todos los clasificadores individuales y se asigna la clase con el máximo valor.

$$\forall patron\ j, \qquad y_j = \arg\max_{i=1,\ldots,C} \left(\sum_{\substack{n=1 \\ \forall n\ asignada}}^{N} f_n^j \right) \tag{3.5}$$

3.1.3.4. Promedio Simple

Para cada patrón se calcula el promedio de las probabilidades de una misma clase que haya sido asignada por todos los clasificadores individuales y se asigna la clase con el máximo valor.

$$\forall patron\ j, \qquad y_j = \arg\max_{i=1,\ldots,C} \left(mean_{\forall n\ asignada} \left(f_1^j, \ldots, f_N^j \right) \right) \tag{3.6}$$

3.1.3.5. Promedio Total

Para cada patrón se calcula el promedio de las probabilidades de una misma clase que haya sido asignada o no por todos los clasificadores individuales, y se asigna la clase con el máximo valor.

$$\forall patron\ j, \qquad y_j = \arg\max_{i=1,\ldots,C} \left(mean_{\forall n} \left(f_1^j, \ldots, f_N^j \right) \right) \tag{3.7}$$

Existen otras reglas u operadores estadísticos como el mínimo o el producto, aunque para el caso de clasificación utilizando las probabilidades a posteriori no son muy recomendables.

3.2. Métodos Propuestos

En esta sección se van a describir los métodos de clasificación propuestos para problemas multiclase con el fin de reducir la carga computacional y temporal que conllevan las técnicas de clasificación basadas en la combinación de clasificadores binarios pareados o *Pairwise*.

En un principio propondremos métodos deconstructivos basados en poda de clasificadores y más tarde se propondrán varios métodos constructivos. En sendas técnicas se utilizarán varias técnicas de eliminación y adición progresivas de clasificadores, y se combinarán las decisiones siguiendo las diversas estrategias de fusión de clasificadores mencionadas en la sección 3.1.

Con las técnicas propuestas en esta sección se realizarán diversos experimentos y se analizarán sus resultados y prestaciones en el Capítulo 4.

3.2.1. Métodos Deconstructivos basados en poda de clasificadores

Comenzamos el estudio realizando métodos de clasificación deconstructivos basados en poda de clasificadores. Estas técnicas se basan en empezar la clasificación utilizando el número máximo de clasificadores pareados posibles para el número de clases de problema e ir reduciendo progresivamente este número. Se calculan, cada vez que se elimina uno de los clasificadores pareados, las prestaciones del clasificador final combinado mediante diversas medidas de evaluación.

En esta sección se van a investigar varias estrategias de combinación de los clasificadores binarios SVM. Se ha probado con los métodos de predicción o combinación de clasificadores: *MaxWins* o voto por mayoría simple propuesto en el artículo para la clasificación *Pairwise, MaxWins Votos* o voto por mayoría ponderada teniendo en cuenta el máximo número de votos posibles, *PPS* o la técnica propuesta en el artículo para el método de clasificación US-MSVM y los métodos de nivel de medidas *Promedio Simple, Promedio Total*, *Suma*, *Mediana* y *Máximo*.

También hemos investigado varios métodos de eliminación de clasificadores para realizar los experimentos. Se ha probado con la eliminación de forma aleatoria, por distancias máximas y por camino “greedy” de mínimo error de clasificación.

3.2.1.1. Método "baseline": Deconstrucción por eliminación de clasificadores pareados de manera Aleatoria

En este apartado se presenta la técnica trivial para resolver este tipo de problemas y cuyos resultados nos servirán para compararlos con los obtenidos mediante los demás métodos propuestos. Lo más sencillo es ir eliminando los clasificadores pareados de forma aleatoria y evaluar las prestaciones de cada estrategia de predicción utilizada. Dada la aleatoriedad de esta técnica se realizan varias iteraciones de este experimento para caracterizar mejor su funcionamiento.

3.2.1.2. Deconstrucción por eliminación de clasificadores pareados basada en Distancias Máximas

Este método de eliminación de clasificadores esta basado en el método US-MSVM que proponen los autores en el artículo y se basa en la estrategia de muestreo de incertidumbre.

Se parte del número total de clasificadores SVM pareados necesario para realizar la clasificación de todas las categorías existentes en el conjunto de datos del problema. Como en el método US-MSVM, se calcula la matriz de decisión DM para dichos clasificadores, siendo cada fila *k* de esta matriz el vector de probabilidades de muestras positivas PPS para cada clase y para el subclasificador pareado *k*. Partiendo de esta matriz, se miden las distancias entre todos los vectores columna y se elige como clasificador a eliminar aquél que esté formado por las clases que estén más distanciadas, es decir, aquellas que sean más distinguibles. Para medir la distancia entre las distintas clases, como se especifica en el artículo, se utiliza la norma 2 o euclídea.

De este modo se realiza el método deconstructivo equivalente al método constructivo publicado en el artículo, ya que este último elegía como siguiente clasificador a entrenar el formado por las clases con mayor incertidumbre o las más indistinguibles.

3.2.1.3. Deconstrucción por eliminación de clasificadores pareados basada en la búsqueda del camino "Greedy" de Error Mínimo de Clasificación

Este método de eliminación se basa en buscar un camino óptimo de mínimo error de clasificación en la fase de test.

Los algoritmos "greedy" son algoritmos que toman decisiones pensando en conseguir resultados en un corto alcance utilizando toda la información de que se dispone y sin pensar en las posibles consecuencias futuras. Son algoritmos eficientes y de fácil implementación y, normalmente, son utilizados para resolver problemas de optimización, en nuestro caso, se va a buscar optimizar el error de clasificación.

Este método tiene como ventajas que es de sencilla elaboración y presenta resultados aceptables. Uno de sus inconvenientes es que tienen un alto coste computacional ya que hay que evaluar muchas combinaciones de clasificadores. Otra desventaja que aparece es que, debido a que se toman decisiones a corto plazo para conseguir los mejores resultados

en el momento, no se garantiza que se obtenga la solución óptima para el problema global incurriendo en mínimos locales, aunque el resultado es bastante bueno. Para encontrar la solución óptima real se pueden realizar métodos por fuerza bruta, que aunque son sencillos de implementar, consumen una gran cantidad de tiempo.

En nuestro problema de clasificación, se parte de todos los clasificadores pareados posibles para el número de clases existentes en los conjuntos de datos, *L*. Se prueban todas las posibles combinaciones de éstos tras eliminar cada clasificador y quedarnos con los *L-1* clasificadores restantes. Se evalúa el error de clasificación de test para cada combinación y nos quedamos con aquella combinación que haya dado el menor error, por tanto, se elimina el clasificador que perturba menos los resultados.

Por ejemplo, para un problema de 4 clases se tienen 6 clasificadores. Entonces, se pueden realizarlas siguientes 6 combinaciones de dichos clasificadores:

	Combinación de Clasificadores					*Error*
elimino 1er Clasif:	C_2	C_3	C_4	C_5	C_6	*0.5*
elimino 2º Clasif:	C_1	C_3	C_4	C_5	C_6	*0.6*
	⋮	⋮	⋮	⋮	⋮	⋮
elimino 6º Clasif:	C_1	C_2	C_3	C_4	C_5	*0.8*

Tabla 3.5: Ejemplo de la combinación de los 6 clasificadores para un problema de 4 clases y su error de clasificación

Puesto que la primera combinación da el menor error, se elimina el 1er clasificador. Para la siguiente iteración nos quedamos con esa combinación de clasificadores y se añade este clasificador al camino "greedy". Se tendrán ahora las siguientes combinaciones:

	Combinación de Clasificadores				*Error*
elimino 2º Clasif:	C_3	C_4	C_5	C_6	*0.95*
elimino 3º Clasif:	C_2	C_4	C_5	C_6	*0.30*
	⋮	⋮	⋮	⋮	⋮
elimino 6º Clasif:	C_2	C_3	C_4	C_5	*0.65*

Tabla 3.6: Ejemplo de la combinación de 5 clasificadores para un problema de 4 clases y su error de clasificación tras haber eliminado un clasificador por error mínimo

Ahora se elimina el tercer clasificador dado que su eliminación produce menor error de clasificación. El camino "greedy" de error mínimo hasta el momento sería C_1-C_3. El algoritmo se repite hasta que se eliminan todos los clasificadores.

3.2.2. Métodos Constructivos

Una vez terminado el estudio de los diversos métodos de deconstrucción de clasificadores, vamos a buscar un método de construcción. Este método es interesante ya que nos permitiría ahorrar coste computacional respecto a los métodos deconstructivos ya que éstos necesitan entrenar el número total de clasificadores pareados.

Los métodos de construcción o de adición de clasificadores pareados que vamos a probar se basan en técnicas equivalentes a las utilizadas en los algoritmos de deconstrucción y se utilizarán las mismas estrategias de combinación de clasificadores o de predicción mencionados en la sección 3.1.

3.2.2.1. Método "baseline": Construcción por adición de clasificadores pareados de manera Aleatoria

Esta es la técnica de construcción trivial que consiste en ir eligiendo cada vez unas determinadas clases de manera aleatoria con las que se entrenará un nuevo clasificador pareado SVM. Progresivamente, se va a ir aumentando el número de clasificadores que se van a combinar para conseguir mejores resultados finales de clasificación. Este método es totalmente equivalente al deconstructivo por lo que sólo hará falta hacer uno de los dos experimentos ya que sus comportamientos serán estadísticamente iguales.

3.2.2.2. Construcción por adición de clasificadores pareados basada en Distancias Mínimas

Esta técnica de construcción es la que proponen los autores en el artículo y que denominan US-MSVM, basada en muestreo de incertidumbre. En ella se van escogiendo las clases con las que se entrenan los clasificadores pareados SVM según la distancia entre ellas, seleccionando aquellas que sean menos distinguibles. Utilizando la matriz de decisión DM, y por tanto las medidas de probabilidad de muestras positivas PPS, se eligen las clases pertenecientes a los vectores columnas de esta matriz con menor distancia, es decir, aquellas con mayor incertidumbre para todos los clasificadores entrenados.

Se realiza por tanto el experimento del artículo pero sin tener en cuenta las restricciones marcadas en éste: la distancia umbral y número máximo de clasificadores a entrenar. De esta manera, se hace un estudio de la influencia del número de clasificadores que hay que entrenar en los resultados finales teniendo en cuenta las diversas estrategias de combinación de clasificadores utilizadas en los demás experimentos.

3.2.2.3. Construcción por adición de clasificadores pareados basada en la búsqueda de un camino de Mínimo Error de Clasificación

Esta técnica de construcción es equivalente al realizar el camino de error mínimo y se basa en ir eligiendo para el entrenamiento los clasificadores formados por pares de clases que producen el error de clasificación mínimo en la fase de test.

Se van a realizar dos técnicas para la elección del siguiente clasificador que se va a añadir, una basada en un algoritmo "greedy" de mínimo error y otra utilizando un matriz de error de clasificación con la que se elige el par de clases con mayor error.

I. Mediante un Algoritmo "Greedy"

Esta técnica es la equivalente al método de deconstrucción por camino "greedy" de mínimo error de clasificación y posee sus mismas propiedades: es sencillo de implementar, presenta resultados aceptables aunque tienen una alta carga computacional y se puede incurrir en mínimos locales.

En este caso se empieza escogiendo aleatoriamente un par de clases con el que se entrenará un clasificador con el que se calcula el error de clasificación para todo el subconjunto de prueba. Para la elección del siguiente par de clases de entrenamiento se prueban todas las combinaciones del clasificador ya entrenado y los *L-1* clasificadores pareados restantes. Se calculan los errores de clasificación de test de cada combinación y se escoge aquella con la que se obtenga el menor valor, es decir, se añade el clasificador que altera menos los resultados. Este proceso se repite hasta que se hayan entrenado todos los clasificadores pareados posibles para las clases del problema, *L*.

Por ejemplo, para un problema de 4 clases se tienen 6 clasificadores pareados posibles. Entonces, si suponemos que se ha entrenado el clasificador C_3 formado por las clases 1 y 4, se pueden realizar las siguientes combinaciones con los 5 clasificadores restantes:

	Combinación de Clasificadores		*Error*
añado 1er Clasif:	C_3	C_1	*0.75*
añado 2º Clasif:	C_3	C_2	*0.8*
	⋮	⋮	⋮
añado 6º Clasif:	C_3	C_6	*0.85*

Tabla 3.7: Ejemplo de la combinación del clasificador C_3 con los 5 clasificadores restantes para un problema de 4 clases y su error de clasificación

Puesto que es la primera combinación con la que se obtiene da el menor error de clasificación, se añade el 1er clasificador. Para la siguiente iteración nos quedamos con esa combinación de clasificadores C_3-C_4 y se prueban ahora las siguientes combinaciones con los 4 clasificadores restantes:

	Combinación de Clasificadores			*Error*
añado 2º Clasif:	C_3	C_1	C_2	*0.7*
añado 4º Clasif:	C_3	C_1	C_4	*0.65*
añado 5º Clasif:	C_3	C_1	C_5	*0.6*
añado 6º Clasif:	C_3	C_1	C_6	*0.75*

Tabla 3.8: Ejemplo de la combinación de los clasificadores C_3- C_1 con los 4 clasificadores restantes para un problema de 4 clases y su error de clasificación

En este caso se añade el quinto clasificador dado que su adición es la que produce el menor error de clasificación y el algoritmo se debe repetir hasta que se añaden todos los clasificadores.

Este método de construcción es completamente equivalente al deconstructivo por lo que su comportamiento estadístico y sus prestaciones serán iguales o muy similares. Como en el caso de la técnica deconstructiva, para hacer las combinaciones de clasificadores se

necesita previamente haber entrenado y evaluado las prestaciones todos los clasificadores pareados. De este modo, esta técnica de construcción no cumple con el objetivo principal de este proyecto ya que no reducirá la carga computacional pero se va a realizar con fines explicativos y comparativos.

II. Mediante una matriz de construcción de error de clasificación

En esta técnica se utiliza una matriz de error de construcción para realizar la elección de los clasificadores a añadir para conseguir un camino de mínimo error de clasificación.

En primer lugar, se eligen un par de clases de forma aleatoria para construir el primer clasificador binario que se va entrenar y se prueban todos los patrones pertenecientes a cada una de las clases existentes en el conjunto de datos. Se miden, utilizando las decisiones obtenidas por el clasificador entrenado para el conjunto de prueba, los errores de clasificación para cada una de las clases.

Para la elección del siguiente par de clases con las que se entrenará el nuevo clasificador SVM, se construye lo que hemos llamado "matriz de construcción". Se define esta matriz como una matriz triangular cuyos elementos representan el error de clasificación para cada par de clases posible del problema. Se rellena esta matriz teniendo en cuenta los resultados del error de clasificación obtenidos y se aumenta el valor a todos los elementos correspondientes a las clases que lo constituyen. Entonces, se busca el elemento de la matriz con el mayor valor y se eligen las clases que lo conforman para entrenar al siguiente clasificador. Para los clasificadores que ya han sido entrenados se cambia el valor del elemento de la matriz correspondiente a cero para que no vuelvan a ser elegidos posteriormente. Se repite todo el proceso hasta que se hayan entrenado y realizado la clasificación con todos los clasificadores.

Con este método se eligen siempre como siguiente par de clases para entrenar el nuevo clasificador aquel que hasta ese momento tiene mayor error de forma conjunta. Se escogen aquellas clases que han tenido mayor error en el anterior paso ya que son aquellas con las que mayor posibilidad de mejora hay en la siguiente iteración del método. De este modo, en el siguiente paso se reducirá el error de clasificación para ambas clases ya que el clasificador combinado final tendrá más aporte de ambas y, por tanto, se reducirá también el error final.

Por ejemplo, para un problema de 6 clases tendríamos una matriz de error de tamaño 5 x 5. Si se ha entrenado el clasificador formado por las clases 4 y 5 (C_4-C_5) y se han obtenido el siguiente vector de errores de clasificación para todas las clases:

Clases	C_1	C_2	C_3	C_4	C_5	C_6
Error (%)	100	100	100	7	10	100

Tabla 3.9: Ejemplo del error de clasificación para todas las clases y para un problema de 6 clases en el que se ha entrenado el clasificador pareado C_4-C_5

Entonces, la matriz de construcción en la que se puede ver el error para cada par de clases y con la que se escogerá aquel par con el que se entrenará el nuevo clasificador en la siguiente iteración quedaría:

	C_1	C_2	C_3	C_4	C_5
C_2	**2**				
C_3	**2**	**2**			
C_4	*1,07*	*1,07*	*1,07*		
C_5	*1,1*	*1,1*	*1,1*	**0**	
C_6	**2**	**2**	**2**	*1,07*	*1,1*

Tabla 3.10: Ejemplo de la matriz de construcción de error para un problema de 6 clases en el que se ha entrenado el clasificador pareado C_4-C_5

Por tanto, se busca en esta matriz el valor máximo del error acumulado por cada par de clases. En este caso existen varios clasificadores posibles para entrenar en la siguiente iteración (C_1-C_2, C_1-C_3, C_1-C_6, C_2-C_3, C_2-C_6 ó C_3-C_6) por lo que se escogerá uno de ellos aleatoriamente.

Imaginemos que se ha escogido aleatoriamente el clasificador C_2-C_3. Se repite el proceso de clasificación, y se obtiene el siguiente nuevo vector de errores de clasificación de test para todas las clases del problema:

Clases	C_1	C_2	C_3	C_4	C_5	C_6
Error (%)	100	10	15	9	14	100

Tabla 3.11: Ejemplo del error de clasificación para todas las clases y para un problema de 6 clases en el que se han entrenado los clasificador pareados C_4-C_5 y C_2-C_3

Ahora tendríamos la siguiente matriz de construcción con los nuevos valores del error para cada par de clases para la siguiente iteración:

	C_1	C_2	C_3	C_4	C_5
C_2	*1,1*				
C_3	*1,15*	**0**			
C_4	*1,09*	*0,19*	*0,24*		
C_5	*1,14*	*0,25*	*0,29*	**0**	
C_6	**2**	*1,1*	*1,15*	*1,09*	*1,14*

Tabla 3.12: Ejemplo de la matriz de construcción de error para un problema de 6 clases en el que se ha entrenado los clasificadores pareados C_4-C_5 y C_2-C_3

En esta iteración se escogería el clasificador C_1-C_6 por tener el mayor error acumulado de todos los pares de clases hasta este momento. Entonces, se habrán entrenado los clasificadores pareados C_4-C_5 y C_2-C_3 con las que se ha conseguido el camino de error mínimo de clasificación.

Por tanto, para esta técnica de construcción se va a conseguir ir reduciendo el error de clasificación final al ir añadiendo clasificadores pareados tras escoger en cada iteración aquel par de clases que posean en ese momento el mayor error acumulado.

Capítulo 4

Trabajo Experimental.

Es este capítulo se presenta el estudio comparativo realizado en este proyecto donde se muestran los diferentes experimentos y sus resultados. Para comenzar se describe las colecciones de datos que van a ser utilizadas. Posteriormente, se detalla cómo se ha realizado la configuración de los conjuntos de datos con los que se han realizado los experimentos propuestos anteriormente y las medidas de evaluación con las que va a hacerse la comparación.

Finalmente se muestran y analizan los resultados obtenidos tras la realización de los experimentos. En la sección anterior hemos presentado dos aproximaciones para reducir la carga computacional de la evaluación de los clasificadores SVM multiclase construidos a partir de combinaciones de clasificadores SVM binarios pareados: una aproximación constructiva y otra aproximación deconstructiva basada en poda de clasificadores. Además, se han presentado varias maneras de combinar estos clasificadores binarios para predecir la clase de cada muestra. Los experimentos presentados en esta sección tienen por objeto evaluar la bondad de estas configuraciones atendiendo a la precisión en la clasificación.

4.1. Colecciones de Datos

Hemos utilizado para los experimentos dos tareas de clasificación: la primera es una clasificación de textos sobre una base de datos de textos del mundo real y la segunda tarea es de reconocimiento de imágenes sobre una colección de dígitos manuscritos.

4.1.1. Bases de Datos de Textos

4.1.1.1. Representación de textos como "bolsas de palabras"

Para representar el lenguaje humano por medio de herramientas informáticas es necesario hacer una representación de dicho lenguaje de forma que pueda ser fácilmente manipulada computacionalmente. Para ello se han desarrollado algunos métodos probabilísticos y estadísticos que han logrado un alto grado de desempeño en la tarea de analizar los documentos.

El inconveniente de éstos métodos es que se basan en la división del documento en palabras o conjuntos de éstas y analizan su importancia dentro del documento basándose únicamente en la frecuencia de aparición (u otros métodos de similitud), sin tener en cuenta su significado ni el significado en relación con el resto de palabras.

Un método para el análisis de textos es el estudio de la semántica de las palabras. La semántica hace referencia al significado que tienen las palabras que componen el texto pero dado que el significado de éstas dependen de otros factores como el orden o el contexto, pueden existir diferentes significados para el mismo conjunto de palabras. Por lo tanto, determinar la semántica del texto puede resultar algo complejo y por esa razón se han propuesto diversos métodos para extraer la mayor cantidad de información y poder aplicarse a varios dominios de aplicación como la clasificación de textos, extracción de resúmenes, entre otros.

En la actualidad existen métodos que han proporcionado buenos resultados y que resultan muy efectivos en cuanto a la exhaustividad y la precisión, y son aquellos basados en la extracción de "bolsas de palabras" (*bags-of-words*). Se fundamentan en encontrar el contexto o tema del documento por medio de la frecuencia de repetición de palabras que lo componen.

En el ámbito de Recuperación de la Información también se utilizan "bolsas de palabras". Cuando un usuario realiza una consulta en un conjunto de documentos, si aumenta el número de palabras comunes entre el documento recuperado (bolsa de palabras) y la consulta, entonces la relación entre ellos es mayor. Por tanto, la RI intenta determinar cuanto se parecen las bolsas de palabras de la consulta y de cada documento.

En la clasificación de documentos, la eficacia de los clasificadores está bastante limitada por como han sido representados los documentos y esta representación generalmente se realiza utilizando la "bolsa de palabras". Por otro lado, en las técnicas de agrupamiento tiene como fin analizar las colecciones y dividir los documentos según su relación y similitud.

Aunque la "bolsa de palabras", como hemos visto, se utiliza bastante en diversos ámbitos ya que genera resultados aceptables de manera fácil y rápida, su principal inconveniente es que no tiene en cuenta la morfología de las palabras (género, número, modo, tiempo, etc.), sus diferentes significados, su sintáctica u orden en la frase y demás variaciones lingüísticas.

4.1.1.2. Colección de Textos 10Newsgroups

En primer lugar, se ha utilizado el corpus *10Newsgroups*, una versión reducida del conjunto de datos *20Newsgroups*[(b)], ya que solamente han sido seleccionadas 10 de las 20 categorías del conjunto original. A su vez, para cada una de las 10 clases escogidas se han seleccionado aleatoriamente 250 documentos para el conjunto de entrenamiento y otros 100 documentos para el conjunto de prueba o test. Las temáticas de las categorías seleccionadas se muestran en la siguiente tabla:

[b] - Se puede descargar la colección *20Newsgroups* en: http://people.csail.mit.edu/jrennie/20Newsgroups/

Grupo	Tema	Grupo	Tema
1	comps.os.ms-windows.misc	6	sci.med
2	rec.sport.baseball	7	talk.politics.misc
3	talk.religion.misc	8	rec.autos
4	sci.space	9	misc.forsale
5	comp.sys.mac.hardware	10	talk.politics.mideast

Tabla 4.1: Temas de las 10 categorías seleccionadas de la colección *10Newsgroups*

Cada documento de esta colección se representa mediante una "bolsa de palabras". En esta representación cada documento se corresponde con un vector de dimensión igual al tamaño del diccionario que se considere. Cada uno de los elementos de ese vector es calculado usando la medida TFIDF (*term frequency-inverse document frequency*) que se usa para evaluar estadísticamente cuan importante es una palabra en el documento. Puede encontrarse más información sobre esta medida en [Salton y Buckley, 1988].

Las categorías de este conjunto están perfectamente balanceadas ya que el número de documentos para cada clase ha sido seleccionado de manera premeditada.

4.1.2. Base de Datos de Imágenes: USPS

En segundo lugar, se ha utilizado el corpus para reconocimiento de dígitos USPS. La tarea de reconocimiento de dígitos es una labor costosa y muy difícil. Esto ocurre ya que algunas muestras son difíciles de clasificar y en que el error estimado que comete una persona puede llegar al 2,5%. A continuación, se puede ver un ejemplo con dígitos difíciles de reconocer:

Figura 4.1: Dígitos difíciles de reconocer en la base de datos *USPS*

El conjunto de datos *USPS*[c] contiene imágenes de dígitos manuscritos en escala de grises que han sido escaneados de los sobres del servicio postal de Estados Unidos. Cada imagen del conjunto tiene una resolución de 16 x 16 píxeles. El conjunto de entrenamiento contiene 7291 imágenes y el de prueba 2007 muestras.

Las categorías de esta colección no están balanceadas, es decir, no están igualmente pobladas. A continuación se muestra una tabla con la población de cada una de las clases:

Dígitos	**"0"**	**"1"**	**"2"**	**"3"**	**"4"**	**"5"**	**"6"**	**"7"**	**"8"**	**"9"**
Población	1553	1269	929	824	852	716	834	792	708	821

Tabla 4.2: Población de cada categoría para USPS

[c] - Se puede descargar la colección en: http://cmp.felk.cvut.cz/cmp/software/stprtool/manual/data/usps2mat.html

4.2. Preprocesado de Datos

Para la realización de los experimentos se debe realizar previamente un procesado de los datos de las colecciones. Lo primero, para cada una de las colecciones y de manera independiente, se debe hacer una normalización de todos los datos para que todos ellos tengan las mismas características y sea más sencilla su comparación. Posteriormente, se debe realizar la división de las colecciones para llevar a cabo los experimentos propuestos.

4.2.1. Normalización de los datos

Es necesario realizar la normalización de los corpus de datos para que todos los documentos o imágenes tengan el mismo peso, cuente como un error o como un acierto.

4.2.1.1. Colección 10Newsgroups

La normalización de la colección de documentos de texto *10Newsgroups* debe hacerse para que cada documento o su "bolsa de palabras" tenga módulo unidad ya que queremos clasificar los textos en función de su contenido. Dado que la representación de los documentos se basa en la frecuencia en la que aparecen las palabras, hay que normalizar la longitud de los textos para que la información que se va utilizar en la clasificación se base exclusivamente en la frecuencia relativa de dichos elementos. De este modo, al normalizar, podemos comparar más fácilmente dos documentos ya que cuanto más similares sean en ambos la frecuencia relativa de las palabras que contiene, más se parecerán los dos documentos.

4.2.1.2. Colección USPS

En la colección de datos *USPS* esta compuesta por imágenes de dígitos en escala de grises. La normalización del conjunto de imágenes de dígitos no debe realizarse de la misma manera que en el caso de documentos. No se puede hacer de este modo ya que normalizar cada imagen para que tenga módulo unidad significaría que se tendría que usar la misma cantidad de tinta para escribir cada dígito y esta situación no ocurre ya que no es lo mismo escribir un 9 que un 1. Por tanto al utilizar la normalización por módulo unidad no sería lo más conveniente en el caso de imágenes ya que se estarían difuminando los tonos más oscuros de los dígitos con más píxeles o los que han sido escritos con un trazo más grueso.

En esta colección, cada imagen tiene una resolución de 16 x 16 píxeles por lo que es representada como un vector de 256 elementos que determinan la posición de cada uno de los 256 píxeles de la imagen. Los valores de los elementos de este vector constituyen un determinado nivel de gris., siendo el valor "0" el nivel de negro y el "256" el nivel de blanco. A continuación se presenta la Figura 4.2 que muestra algunos ejemplos de las imágenes que representan dígitos manuscritos del conjunto *USPS*:

Figura 4.2: Imágenes en escala de grises de algunos dígitos de la colección *USPS*

Para concluir debemos representar todas las imágenes para que todas tengan el mismo peso. Cada elemento del vector de píxeles, o su posición en la imagen, tendrá un valor entre 0 y 256 dependiendo si hay en esa posición se ha escrito el dígito y/o la cantidad de tinta del trazo, todas las imágenes se comportan de la misma manera. Por esta razón, la normalización de este conjunto es muy sencilla y se basa en hacer que los valores que determinan los niveles de gris del vector de píxeles se encuentren entre 0 y 1.

4.2.2. División de las colecciones

Todos los documentos de las colecciones utilizadas están previamente etiquetados, por lo que es sencillo proceder a su división. Cada una de las colecciones se ha dividido en dos conjuntos, uno de entrenamiento que sirve para entrenar el clasificador, y otra de prueba o test, que sirve para evaluar las prestaciones del sistema en datos no usados para entrenarlo.

Esta división, para la colección de textos *10Newsgroups*, como se ha explicado anteriormente, se ha hecho de manera explícita seleccionando 250 documentos de cada clase para el conjunto de entrenamiento y 100 documentos de cada clase para el conjunto de test. En cambio la división para la colección de imágenes *USPS* se ha hecho en un porcentaje del 80% para el conjunto de entrenamiento y del 20% para el conjunto de test.

4.3. Medidas de Evaluación

Es necesario escoger una medida para evaluar el rendimiento de los algoritmos propuestos y poder de esta manera comparar sus prestaciones.

A continuación, se describen algunas medidas de evaluación que se utilizan comúnmente en el área de la clasificación y en la recuperación de información. Estas medidas de evaluación nos dan información acerca de cuan de efectivos son los sistemas de clasificación.

Para el caso de la evaluación de un sistema de clasificación se basa, comúnmente, en el cálculo de la matriz de confusión. Esta matriz nos da información sobre el comportamiento del sistema de clasificación y como éste se “confunde” en sus predicciones.

A continuación, se muestra la tabla de contingencia o matriz de confusión para cada clase *i* de un problema multiclase:

Para *clase i*	**Valores Predichos**	
Valores Reales	TN_i	FP_i
	FN_i	TP_i

Tabla 4.3: Matriz de confusión para un problema multiclase para la clase i

Las filas de la matriz representan los valores reales mientras que las columnas representan los valores de predicción del modelo. La matriz se crea ordenando todos los casos en varios estados: si el valor predicho coincide con el valor real, y si este valor era correcto o incorrecto. Estos estados se conocen como verdaderos positivos (TP_i), falsos positivos (FP_i), verdaderos negativos (TN_i) y falsos negativos (FN_i) para la clase *i*. Siendo, para la clase *i*:

$TP_i \equiv$ Número de predicciones correctas cuando la instancia es positiva
$FP_i \equiv$ Número de predicciones incorrectas cuando la instancia es positiva
$TN_i \equiv$ Número de predicciones correctas cuando la instancia es negativa
$FN_i \equiv$ Número de predicciones incorrectas cuando la instancia es negativa

En el caso de los sistemas de recuperación de la información estas medidas nos dan información de la relevancia de los documentos recuperados tras una consulta del usuario y también de la relación estadística existente entre los conjuntos de documentos.

En el siguiente dibujo podemos ver los grupos en los que se divide la colección de documentos en función de si han sido recuperados y de su relevancia.

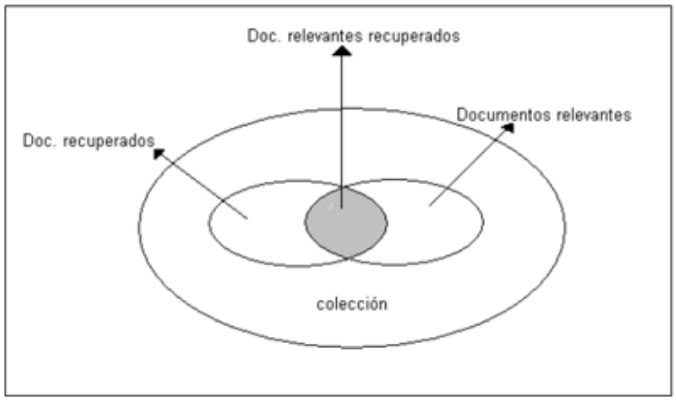


Figura 4.3: División de la colección de documentos para sistemas de recuperación de la información

En el cálculo de la efectividad y eficiencia de un sistema se utilizan varias medidas como la precisión, la exhaustividad, la medida F, la exactitud y el error de clasificación. Se calculan estas medidas para cada una de las clases del problema.

4.3.1. Precisión

En sistemas de clasificación, la precisión de la clase *i* es la proporción de patrones predichos positivamente que son correctamente clasificados:

$$precision_i = \frac{TP_i}{TP_i + FP_i} \tag{4.1}$$

En sistemas de recuperación de la información, la precisión es la proporción de documentos con información realmente relevante para el usuario del total de los documentos recuperados:

$$precision = \frac{Documentos\ Relevantes\ Recuperados}{Documentos\ Recuperados} \tag{4.2}$$

4.3.2. Exhaustividad

En sistemas de clasificación, para la clase *i*, la exhaustividad o *recall* es la proporción de patrones predichos positivamente que son correctamente identificados:

$$recall_i = \frac{TP_i}{TP_i + FN_i} \tag{4.3}$$

En sistemas de recuperación de la información, la exhaustividad es la proporción de documentos recuperados con información relevante, del total de documentos relevantes de la colección:

$$recall = \frac{Documentos\ Relevantes\ Recuperados}{Documentos\ Relevantes} \tag{4.4}$$

4.3.3. Medida F

Una de las medidas más utilizadas para evaluar la efectividad de los diferentes clasificadores o en recuperación de información es la medida F_β que tiene en consideración las dos medidas mencionadas anteriormente, la precisión y la exhaustividad. Podría decirse que es una medida ponderada de ambas:

$$F_\beta = (1+\beta^2)\frac{precision \cdot recall}{\beta^2\ precision + recall} \tag{4.5}$$

siendo $\beta=1$, el peso más común y por tanto, F_1 la medida más utilizada:

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} \tag{4.6}$$

y sustituyendo se obtiene para la clase *i*:

$$F_{1,i} = 2\frac{TP_i}{2\ TP_i + FP_i + FN_i} \tag{4.7}$$

4.3.4. Exactitud y Error

La exactitud o *accuracy* mide la proporción de patrones que han sido bien predichos en relación al número total de éstos. Esta medida da una idea de la fiabilidad del sistema y está íntimamente ligada al error ya que ambos valores suman 1. Para cada clase *i* del problema multiclase tenemos:

$$accuracy = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, \tag{4.8}$$

por tanto:

$$error = 1 - accuracy = \frac{FP_i + FN_i}{TP_i + TN_i + FP_i + FN_i} \tag{4.9}$$

4.3.5. Cálculo de las medidas globales

Las fórmulas recogidas anteriormente son medidas para una única clase. Dado que la mayoría de los conjuntos de documentos pueden clasificarse también en varias categorías o temas, debemos utilizar unas medidas que promedien tanto las clases como el número de documentos existentes. Para obtener los valores globales para analizar el experimento podemos utilizar dos diferentes enfoques:

4.3.5.1. Micro-averaging o Micropromedio

Este método calcula el promedio de las medidas definidas por las suma de todos los valores individuales de TP_i, FP_i, TN_i y FN_i para cada clase o documento.

Las fórmulas de precisión y recall son:

$$precision^m = \frac{\sum_i \sum_j TP_{ij}}{\sum_i \sum_j TP_{ij} + \sum_i \sum_j FP_{ij}} \tag{4.10}$$

$$recall^m = \frac{\sum_i \sum_j TP_{ij}}{\sum_i \sum_j TP_{ij} + \sum_i \sum_j FN_{ij}} \tag{4.11}$$

donde TP_{ij}, FP_{ij}, y FN_{ij} son el número de verdaderos positivos, falsos positivos y falsos negativos encontrados para la clase *i* en la evaluación del documento *j*.

4.3.5.2. Macro-averaging o Macropromedio

En esta técnica se calculan primero las sumas locales de las medidas TP_i, FP_i, TN_i y FN_i para cada categoría *i* evaluándolas en cada uno de los documentos *j* de la colección, obteniendo después la media global. Entonces, las fórmulas de precisión y recall son:

$$precision_j = \frac{\sum_i TP_{ij}}{\sum_i TP_{ij} + \sum_i FP_{ij}} \quad \text{y} \quad recall_j = \frac{\sum_i TP_{ij}}{\sum_i TP_{ij} + \sum_i FN_{ij}} \qquad (4.12 \text{ y } 4.13)$$

$$precision^M = \frac{\sum_j precision_j}{n} \quad \text{y} \quad recall^M = \frac{\sum_j recall_j}{n} \qquad (4.14 \text{ y } 4.15)$$

donde *n* es el número total de categorías en las que puede clasificarse la colección de documentos.

Partiendo de estas medidas promediadas tenemos las siguientes fórmulas globales del F_1, la exactitud o *accuracy* y del error:

$$F_{1j} = \frac{2\sum_i TP_{ij}}{2\sum_i TP_{ij} + \sum_i FP_{ij} + \sum_i FN_{ij}} \rightarrow F_1^M = \frac{\sum_j F1_j}{n} \qquad (4.16 \text{ y } 4.17)$$

$$accuracy_j = \frac{\sum_i TP_{ij} + \sum_i TN_{ij}}{\sum_i TP_{ij} + \sum_i TN_{ij} + \sum_i FP_{ij} + \sum_i FN_{ij}} \qquad (4.18)$$

$$error_j = 1 - accuracy_j = \frac{\sum_i FP_{ij} + \sum_i FN_{ij}}{\sum_i TP_{ij} + \sum_i TN_{ij} + \sum_i FP_{ij} + \sum_i FN_{ij}} \qquad (4.19)$$

Para las ecuaciones del error y del *accuracy*, dado el hecho de que los valores de los FP calculados para una determinada clase provocan resultados de FN y TN en las restantes clases, podemos eliminar elementos redundantes, quedando las ecuaciones globales de la siguiente manera:

$$accuracy_j = \frac{\sum_i TP_{ij}}{\sum_i TP_{ij} + \sum_i FP_{ij}} \rightarrow accuracy^M = \frac{\sum_j accuracy_j}{n} \qquad (4.20 \text{ y } 4.21)$$

$$error_j = 1 - accuracy_j = \frac{\sum_i FP_{ij}}{\sum_i TP_{ij} + \sum_i FP_{ij}} \rightarrow error^M = \frac{\sum_j error_j}{n} \qquad (4.22 \text{ y } 4.23)$$

En nuestro proyecto hemos elegido como medidas de evaluación de los métodos propuestos el error y la medida F_1 para poder comparar las prestaciones y el buen comportamiento de los mismos.

4.4. Guardado de Información: Reducción del Coste Computacional

Lo primero que se ha hecho es realizar las etapas de entrenamiento y de prueba para todos los clasificadores posibles y ambos conjuntos de datos, con el fin de poder salvar los resultados. De este modo, ya que solamente se deberá hacer el entrenamiento y el test una única vez, se reduce notablemente la complejidad y el tiempo en las simulaciones.

En ambos corpus se tienen 10 categorías, para *10Newsgroups* las clases son las que se pueden ver en la Tabla 4.1 y para *USPS* son los dígitos del 0 al 9.

Caso Pairwise

El número de clasificadores pareados (*1-vs-1*) que se tienen que entrenar para nuestro problema es:

$$\#Clasificadores\_Pairwise = \frac{N\ (N-1)}{2}\ , \tag{4.1}$$

con N=10, $$\#Clasificadores\_Pairwise = 45 \tag{4.2}$$

Caso 1-vs-All

El número de clasificadores *1-vs-resto* que se entrenan es igual al número de clases del problema, por tanto:

$$\#Clasificadores\_1vsAll = 10 \tag{4.3}$$

Después del entrenamiento y del test, se guarda la información para cada patrón de prueba y para cada uno de los clasificadores. Entonces para cada clasificador formado por el par de clases C_i y C_j, se guardan para todas las muestras de test los vectores de clase predicha (C_i o C_j), los vectores de decisión (+1 ó -1), los vectores de probabilidades estimadas a posteriori (probabilidad de decidir C_i o C_j) y la matriz de decisión DM.

Para el clasificador [C_i, C_j], se guardan los siguientes vectores:

	Clase Pred	*Decisión*	*Prob (clase)*
patrón 1	C_i	+1	0.95 (C_i)
patrón 2	C_i	+1	0.80 (C_i)
patrón 3	C_j	-1	0.65 (C_j)
□	⋮	⋮	⋮
patrón k	C_i	+1	0.70 (C_i)
□	⋮	⋮	⋮
patrón K	C_j	-1	0.86 (C_j)

Tabla 4.4: Ejemplo de la información que es guardada para reducir la complejidad y el tiempo

El hecho de guardar esta información, como ya se ha comentado, va a reducir la complejidad y el tiempo que conllevan las simulaciones de los experimentos que se van a realizar. En el caso *1-vs-All* la utilización de los datos recopilados va a resultar beneficioso ya que, aunque hay que entrenar un menor número de clasificadores que en el caso *Pairwise*, estos deben utilizar un elevado número de muestras por lo que normalmente conllevaría un elevado coste temporal.

4.5. Presentación de Resultados Experimentales

Nos hemos servido de los clasificadores binarios de máquinas de vector soporte basados en regularización (C-SVM) obtenidos de la librería *LIBSVM* y, para realizar dichos experimentos, se ha utilizado la herramienta de software matemático Matlab 2008a.

A continuación se van a presentar los resultados de los métodos de clasificación descritos anteriormente.

4.5.1. Caso 1-vs-All

En primer lugar, se va a realizar el experimento de clasificación 1-vs-Todos (*1-vs-All*) en la que se han entrenado tantos clasificadores como el número de clases del problema.

Se han evaluado las prestaciones de este método midiendo el error de clasificación y la medida F_1, para poder después compararlos con los obtenidos en el resto de experimentos propuestos.

El error de clasificación de test para este experimento es del 9.70% para el conjunto de datos *10Newsgroups* y del 5.08% para el conjunto *USPS*.

GRUPOS	Grp1	Grp2	Grp3	Grp4	Grp5	Grp6	Grp7	Grp8	Grp9	Grp10
1-vs-All (simulación)	90,10%	93,07%	93,47%	94,00%	84,06%	90,29%	87,63%	88,54%	87,63%	94,12%

Tabla 4.5: Medida F_1 obtenida con el experimento *1-vs-All para* el conjunto *10Newsgroups*

GRUPOS	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
1-vs-All (simulación)	97,25%	97,69%	92,27%	93,54%	92,38%	92,35%	95,52%	95,83%	93,62%	95,48%

Tabla 4.6: Medida F_1 obtenida con el experimento *1-vs-All* para el conjunto *USPS*

4.5.2. Replicar los resultados del experimento Pairwise del artículo

Como paso previo se va a intentar replicar los resultados que se obtienen en el artículo para el caso del método *1-vs-1* o *pairwise* para SVM multiclase.

En el artículo se dice que en la fase de test del caso *pairwise* se adopta el algoritmo MaxWins o de votación por mayoría simple para decidir la categoría final tras la combinación de todos los clasificadores binarios SVM, por lo que en nuestra simulación se utilizará esta técnica de decisión.

En las siguientes tablas (4.7 y 4.8), se muestran la comparación de las medidas F_1 publicadas en el artículo y las obtenidas tras la simulación de nuestro experimento para las dos colecciones de datos en uso.

GRUPOS	Grp1	Grp2	Grp3	Grp4	Grp5	Grp6	Grp7	Grp8	Grp9	Grp10
Pairwise (artículo)	87,5%	95,4%	81,2%	90,7%	85,0%	88,5%	73,0%	93,3%	80,2%	87,2%
Pairwise (simulación)	87,44%	90,29%	93,33%	92,54%	85,58%	89,76%	90,00%	89,01%	85,13%	95,00%

Tabla 4.7: Comparación de la F1 obtenida con el experimento pairwise para *10Newsgroups*

GRUPOS	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
Pairwise (artículo)	93,5%	96,2%	88,6%	90,0%	95,6%	89,3%	96,0%	92,4%	91,5%	95,5%
Pairwise (simulación)	98,20%	97,89%	91,41%	93,29%	93,33%	92,64%	94,40%	96,53%	92,73%	95,56%

Tabla 4.8: Comparación de la F1 obtenida con el experimento pairwise para USPS

Como se puede observar los resultados obtenidos en nuestra simulación mejoran los obtenidos por los autores en el artículo para casi todas las clases. En el caso del conjunto de datos *10Newsgroups* incluso mejoramos la medida F_1 en un 17% para los patrones del Grupo7 y en el conjunto *USPS* mejoramos hasta en un 4.7% para el dígito "0".

4.5.3. Replicar los resultados de la estrategia US-MSVM

En primer lugar, se va a realizar una simulación del experimento propuesto en el artículo para intentar replicar los resultados que los autores obtuvieron.

Siguiendo las indicaciones de los autores, utilizamos clasificadores SVM fijando el parámetro de regularización C=100 y con una función de kernel tipo RBF con $\gamma=0.1$.
Como parámetros del método US-MSVM se eligieron, como se usa en el artículo, la distancia umbral $d^*=0.8$, el número máximo de clasificadores r=23 y el tamaño del subconjunto de entrenamiento de 30 patrones por clase. Dado el carácter aleatorio de este método con la elección del primer par de clases de entrenamiento, hemos realizado 50 simulaciones calculando los resultados medios para lograr una mejor caracterización del modelo.

GRUPOS	Grp1	Grp2	Grp3	Grp4	Grp5	Grp6	Grp7	Grp8	Grp9	Grp10
US-MSVM (artículo)	87,5%	93,8%	81,2%	86,9%	84,5%	86,1%	73,0%	91,4%	78,6%	87,2%
US-MSVM (simulación)	66,36%	48,12%	67,42%	44,79%	64,08%	49,48%	60,84%	53,64%	64,88%	85,86%

Tabla 4.9: Comparación de la F1 obtenida con el experimento *US-MSVM* para *10Newsgroups*

GRUPOS	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
US-MSVM (artículo)	93,5%	95,8%	88,0%	89,1%	93,1%	86,9%	95,5%	91,7%	90,3%	94,2%
US-MSVM (simulación)	91,09%	45,81%	64,53%	74,01%	71,28%	73,94%	75,13%	63,96%	82,84%	85,05%

Tabla 4.10: Comparación de la F_1 obtenida con el experimento *US-MSVM* para *USPS*

Los resultados logrados en la simulación difieren mucho de los publicados en el artículo y se puede observar que son mucho peores. Para el caso del corpus *10Newsgroups* se alcanzan diferencias de hasta casi del 50% para el Grupo2 y de más del 25% de media para todas las clases. De igual manera ocurre para el caso de *USPS* ya que los resultados difieren en casi el 50% para el dígito "1" y en casi el 20% de media. En este caso no se ha conseguido replicar los resultados publicados del método US-MSVM aunque se siguieron todas las indicaciones y restricciones que los autores especificaban en él.

Tras realizar estos tres experimentos basados en multiclasificación SVM, en nuestro proyecto vamos investigar otras técnicas de clasificación multiclase basadas en SVM con las que intentaremos conseguir una reducción del número de clasificadores que es necesario entrenar y, de este modo, disminuir el coste computacional y temporal.

4.5.4. Métodos Deconstructivos basados en poda de Clasificadores

Comenzamos realizando experimentos de los métodos de clasificación deconstructivos descritos en la sección 3.2.1 de este proyecto donde se hará una comparación de las técnicas de eliminación propuestas. Para evaluar los resultados de la clasificación se van a usar varias estrategias de combinación de los clasificadores binarios.

4.5.4.1. Método "baseline": Deconstrucción por eliminación de clasificadores pareados de manera Aleatoria

Empezamos con el experimento más simple que es ir eliminando los clasificadores binarios de forma aleatoria y evaluando las prestaciones de cada estrategia de predicción utilizada. Para caracterizar mejor el funcionamiento de esta técnica aleatoria se realizan 10 iteraciones de este experimento.

En la gráficas, se muestra una evolución del error de clasificación de test según el número de clasificadores entrenados para este método de eliminación y utilizando varias estrategias de combinación para el conjuntos de datos *10Newsgroups* y *USPS*.

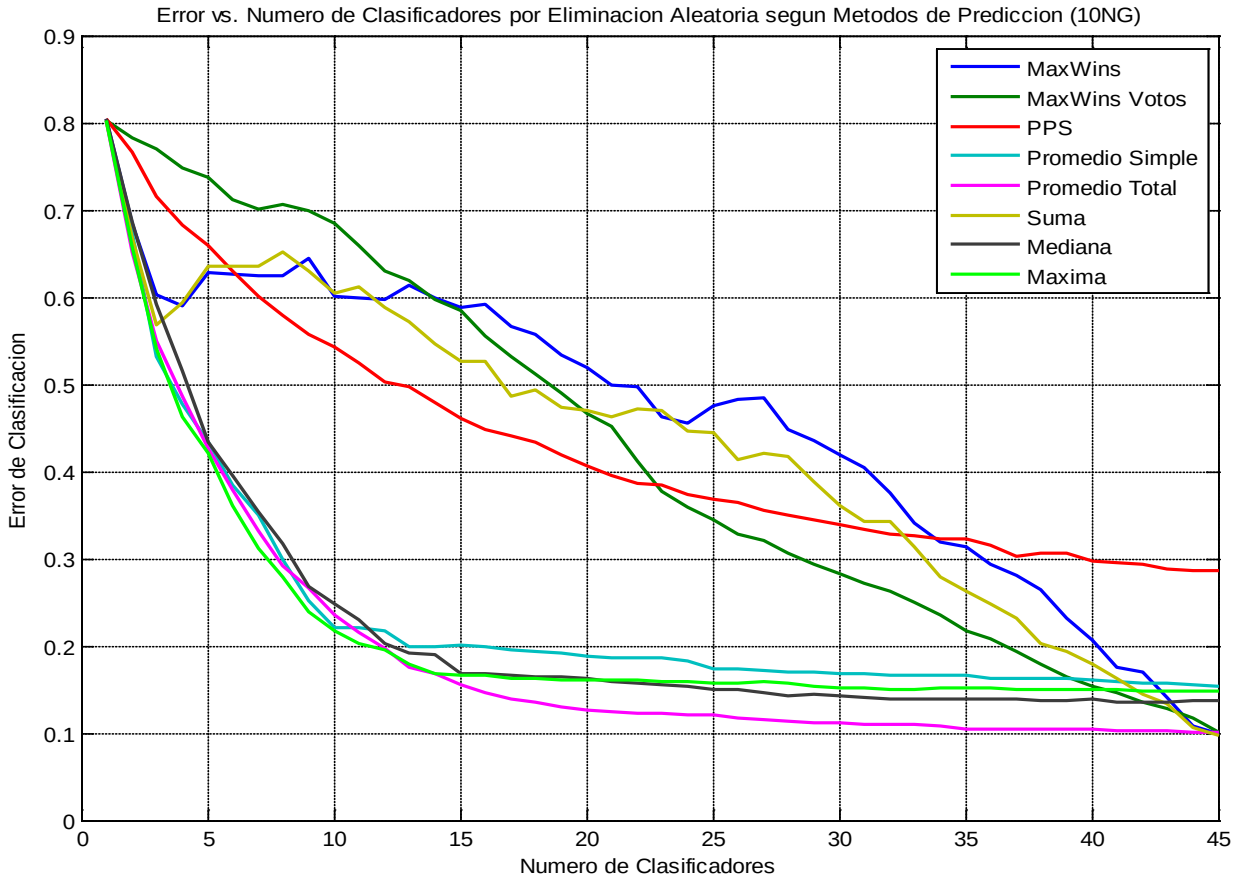


Figura 4.4: Evolución del error de clasificación en función del número de clasificadores entrenados para la técnica de eliminación aleatoria y varias estrategias de predicción para *10Newsgroups*

Para el primer conjunto de datos, al comparar los métodos propuestos en el artículo podemos concluir que *PPS* da mejores resultados para un número bajo de clasificadores, hasta cerca de la mitad del total posible, para el caso de utilizar *MaxWins Votos* y para la técnica *MaxWins* clásica es mejor en un rango entre 5 y 33 clasificadores. Estos resultados pueden justificarse ya que la técnica *PPS* a partir de un número de clasificadores empieza a entrenar clasificadores que no aportan mucha información ya que son los primeros que se entrenan los más "útiles", en cambio, en las técnicas de voto por mayoría se obtienen mejores resultados cuanto mayor sea el número de clasificadores puesto que cada clase recibirá un mayor número de votos y será más probable que se decida la categoría real. Para terminar con este conjunto se puede observar que las estrategias de combinación basadas en métodos de nivel de confianza, excepto la *Suma* que sigue el mismo comportamiento que *MaxWins*, dan resultados mejores que los propuestos en el artículo, *PPS*, para todo el rango de clasificadores entrenados, siendo el método *Promedio Total* el que da los mejores resultados.

Si analizamos los resultados para el conjunto de datos *USPS* y comparamos la técnicas *PPS* y las de votos por mayoría, también es mejor para un número bajo de clasificadores pero este número aumenta con respecto al otro grupo de datos, siendo mejor hasta unos 32 para *MaxWins Votos* y hasta unos 38 para *MaxWins*. Para este conjunto los métodos de nivel de confianza también se comportan bastante bien siendo la técnica de *Promedio Total* con la que se siguen consiguiendo los mejores resultados para cualquier número de clasificadores pareados entrenados.

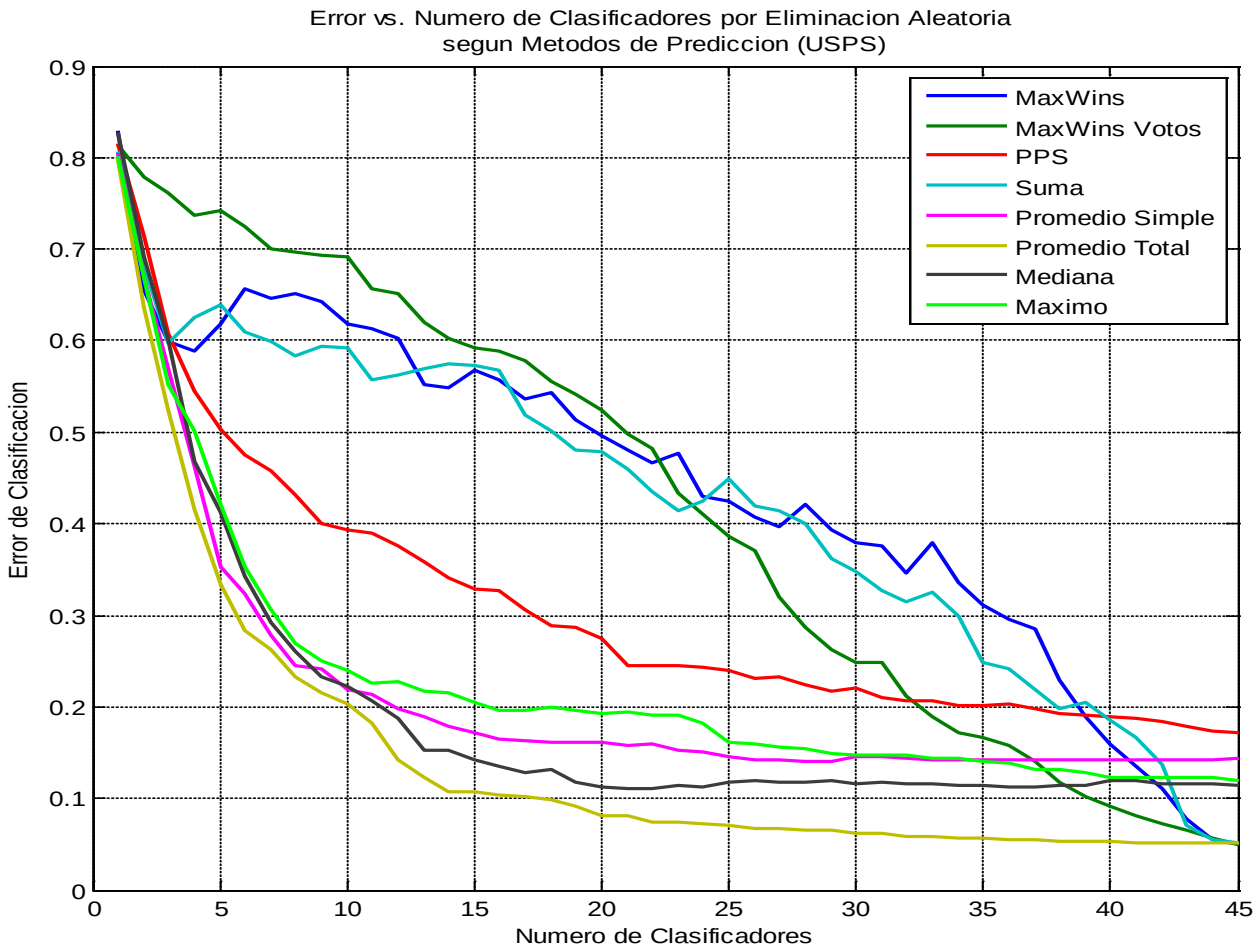


Figura 4.5: Evolución del error de clasificación en función del número de clasificadores entrenados para la técnica de eliminación aleatoria y varias estrategias de predicción para *USPS*

4.5.4.2. Deconstrucción por eliminación de clasificadores pareados basada en Distancias Máximas

Este método de eliminación de clasificadores, como ya se ha comentado, esta basado en la idea que proponen los autores en el artículo y en estrategias de muestreo de incertidumbre.

Se selecciona el clasificador entrenado cuyas clases tengan mayor distancia para ser eliminado en la siguiente iteración.

En las gráficas de evolución del error de clasificación de test en función del número de clasificadores entrenados y los métodos de predicción, podemos sacar las mismas conclusiones que en el caso aleatorio.

En el caso del conjunto *10Newsgroups*, los métodos de predicción por medidas de nivel dan muy buenos resultados siendo el *Promedio Total* la mejor de estas técnicas. Comparando los casos de *MaxWins* o *MaxWins ponderado* con la técnica propuesta en el artículo *PPS*, esta última es mejor para un número de clasificadores menor de 40 y 23 respectivamente.

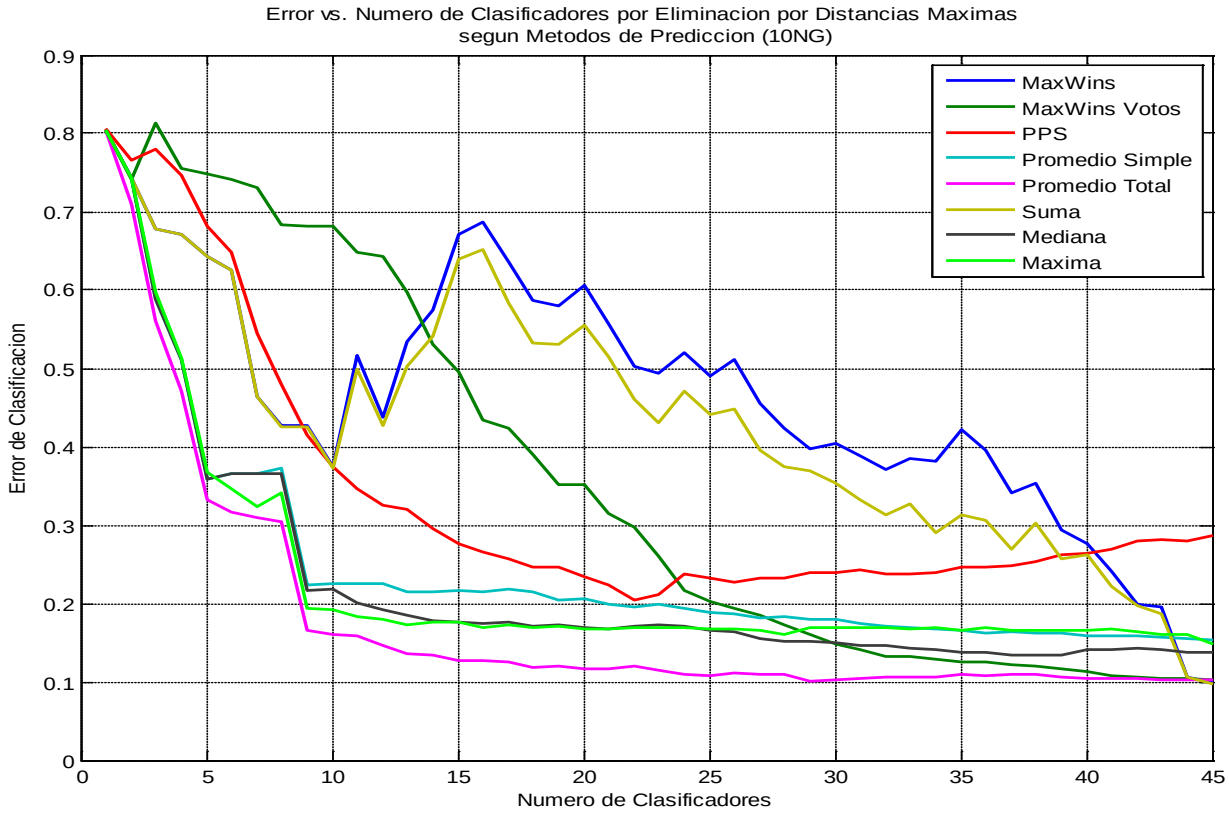


Figura 4.6: Evolución del error de clasificación en función del número de clasificadores entrenados para la técnica de eliminación por distancias máximas y varias estrategias de predicción para *10Newsgroups*

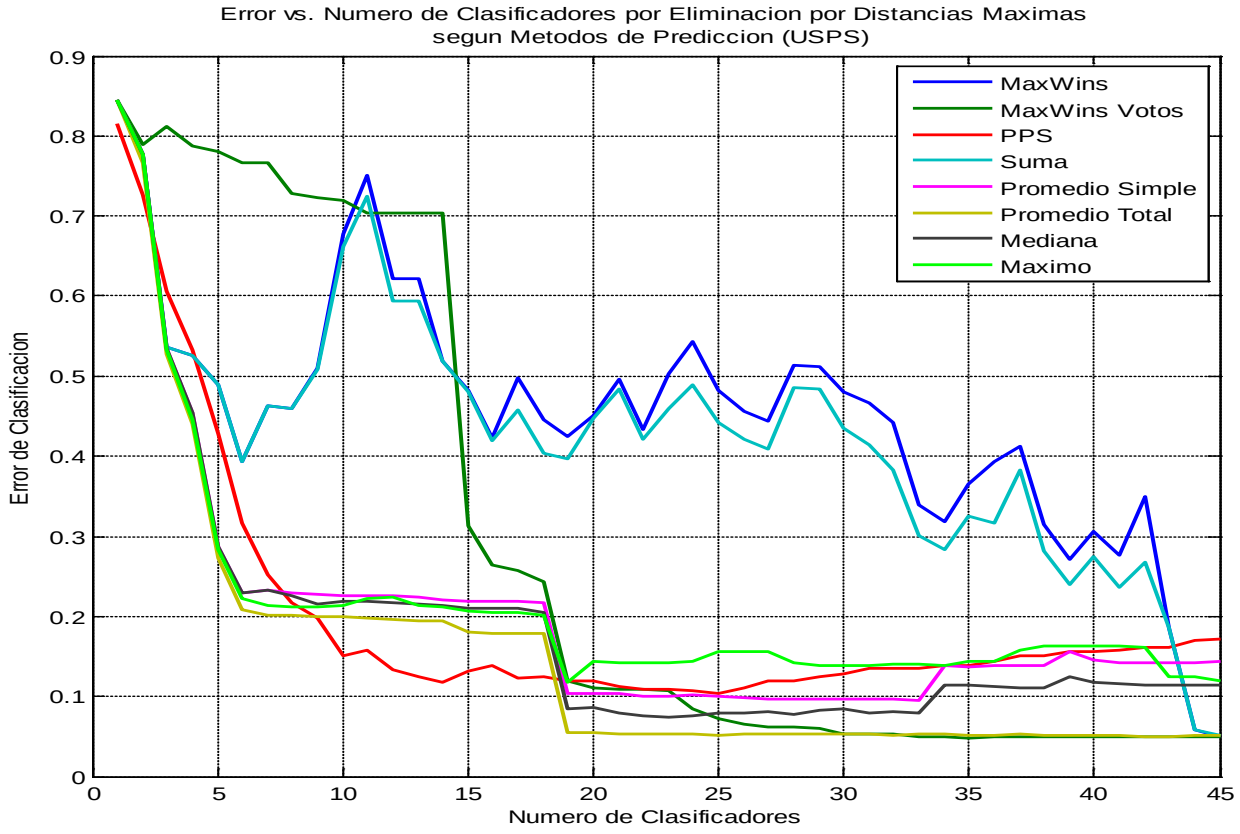


Figura 4.7: Evolución del error de clasificación en función del número de clasificadores entrenados para la técnica de eliminación por distancias máximas y varias estrategias de predicción para *USPS*

4.5.4.3. Deconstrucción por eliminación de clasificadores pareados basada en la búsqueda del camino "Greedy" de Error Mínimo de Clasificación

Este método de eliminación se basa en el algoritmo de optimización llamado "greedy", en el que buscaremos un camino "óptimo" de mínimo error de clasificación. En realidad se alcanzará un resultado subóptimo ya que este tipo de métodos intentan mejorar las prestaciones en un corto alcance sin pensar mucho en lo que provocarán las decisiones actuales en los resultados futuros. Pese a encontrar un camino subóptimo dan resultados bastante mejores que otras técnicas y se va a utilizar ya que es de fácil implementación aunque de alto coste computacional.

En las gráficas se muestra la evolución del error de clasificación según el número de clasificadores que han sido entrenados para varias técnicas de combinación final de clasificadores.

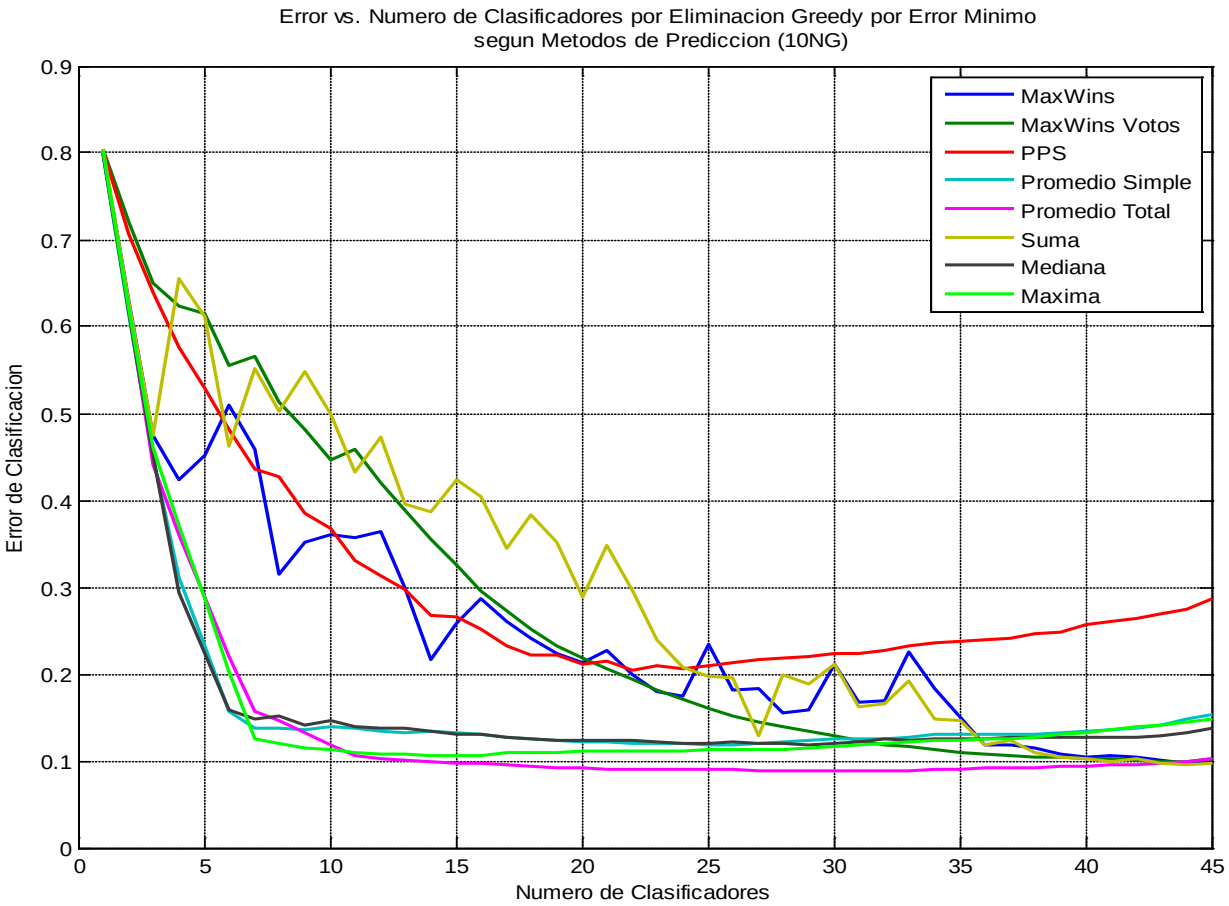


Figura 4.8: Evolución del error de clasificación en función del número de clasificadores entrenados para la técnica de eliminación por camino "greedy" de mínimo error y varias estrategias de predicción para *10Newsgroups*

Para el conjunto de datos *10Newsgroups* las técnicas de predicción final utilizando reglas estadísticas fijas dan los mejores resultados, siendo de nuevo la basada en *Promedio Total* la mejor de todas ellas. Para este corpus de datos y en este caso de optimización por camino "greedy", el método *MaxWins* para un bajo número de clasificadores tiene un comportamiento similar al *PPS* pero es más inestable. *MaxWins* tiene un comportamiento

zigzagueante que alterna el valor del error dependiendo del número de clasificadores. Este comportamiento es debido a que la eliminación de varios clasificadores que deciden sobre la misma clase hace que ésta reciba menor número de votos haciendo que su influencia sobre la decisión final sea menor, produciéndose así mayor número de errores hasta que otra vez el número de votos de la clase correcta sea decisivo. En este caso puede verse claramente que el uso de la técnica *MaxWins Votos*, basada en voto por mayoría ponderada en la que se tienen en cuenta todos los votos posibles para cada clase, suaviza el comportamiento de la curva con respecto a *MaxWins* pero sólo es mejor a *PPS* a partir de 20 clasificadores aunque a partir de ese punto el error disminuye considerablemente.

En el caso del conjunto de datos *USPS*, siguen siendo las técnicas basadas el reglas fijas las que mejor comportamiento tienen ya que mejoran el valor de error logrado en *PPS* para todo el rango de clasificadores y es la de *Promedio Total* la mejor de ellas. La técnica *MaxWins Votos* mejora el error obtenido con *PPS* a partir de unos 18 clasificadores y alcanza los valores de error para la técnica *Promedio Total* a partir de 27.

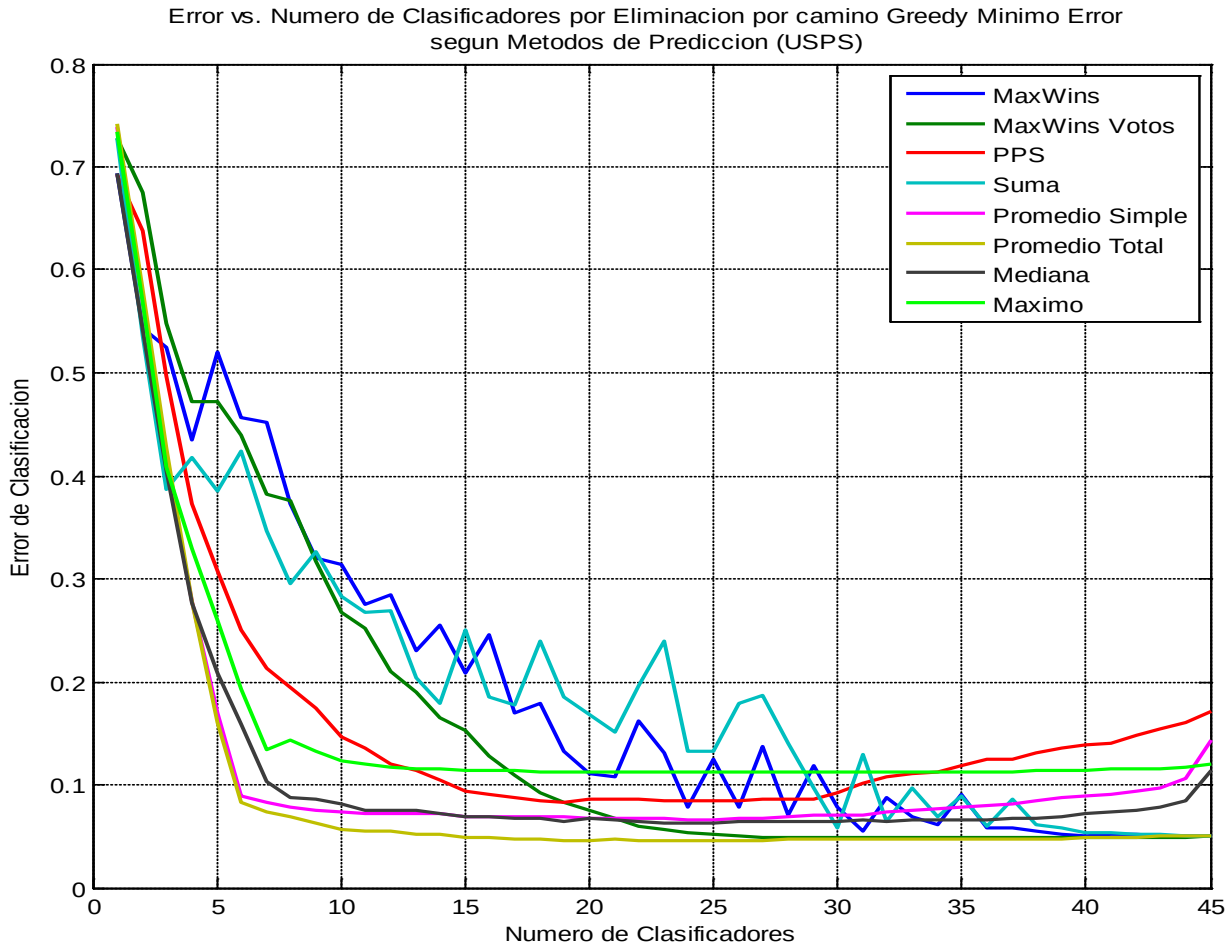


Figura 4.9: Evolución del error de clasificación en función del número de clasificadores entrenados para la técnica de eliminación por camino "greedy" de mínimo error y varias estrategias de predicción para *USPS*

4.5.4.4. Comparación de la Técnicas de Deconstrucción

A continuación, para los dos conjuntos de prueba *10Newsgroups* y *USPS*, se hace una comparación de las tres técnicas de eliminación utilizadas basadas en clasificadores pareados o *pairwise* y la técnica de clasificación basada en clasificadores *1-vs-All*. Se van a comparar dichas técnicas de deconstrucción en las que se ha aplicado la mejor estrategia de predicción o combinación *Promedio Total*.

Se muestra gráficamente la evolución del error dependiendo del número de clasificadores y las técnicas de eliminación utilizadas mostrando para cada una de ellas, únicamente, la curva de la estrategia de predicción con la se obtuvieron los mejores resultados.

Para el primer conjunto de datos *10Newsgroups* la técnica de eliminación por camino "greedy" de error mínimo la mejor de ellas. Con esta última técnica incluso se obtiene resultados mejores que utilizando la clasificación *1-vs-All* a partir de unos 15 clasificadores.

Para el segundo conjunto *USPS*, también la mejor técnica de clasificación para todo el rango de clasificadores es la de eliminación por camino "greedy" de mínimo error, también consiguiendo errores menores que los obtenidos con *1-vs-All* a partir de 15 clasificadores.

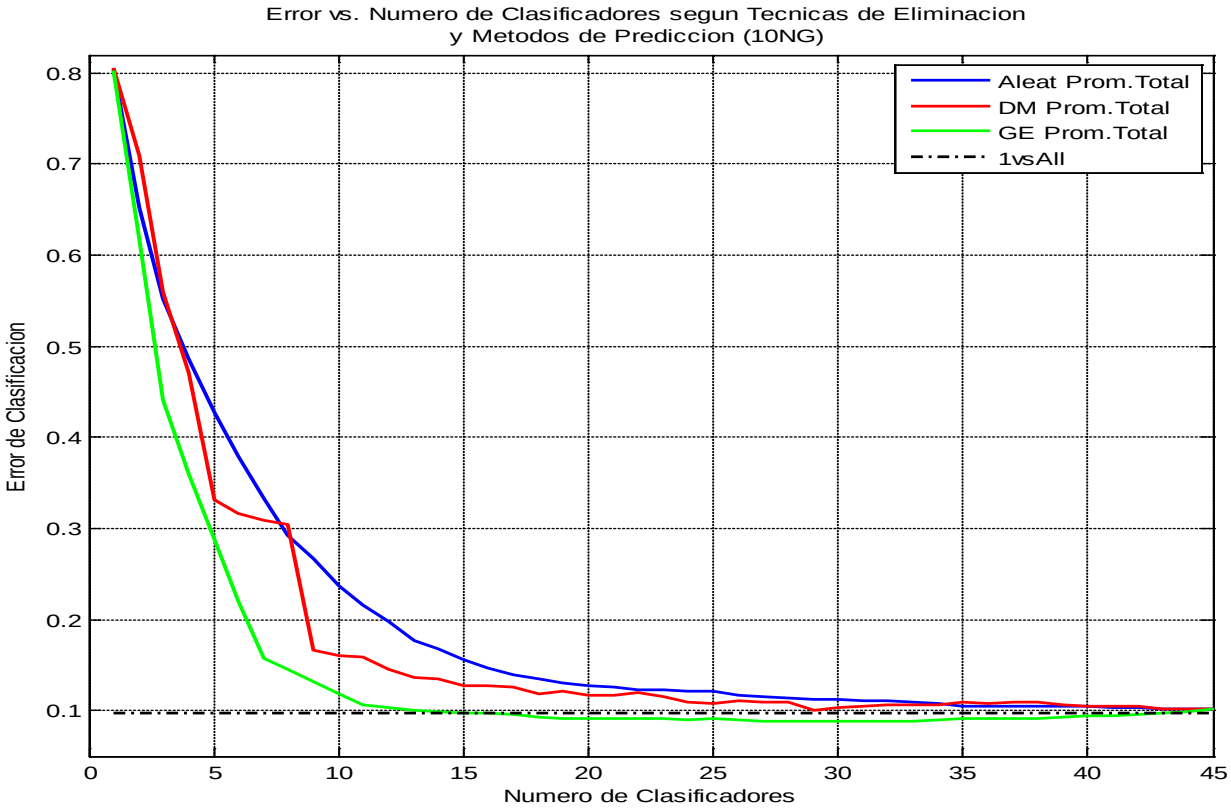


Figura 4.10: Comparación de la evolución del error de clasificación en función del número de clasificadores de las tres técnicas de eliminación y la mejor estrategia de predicción para cada una para *10Newsgroups*

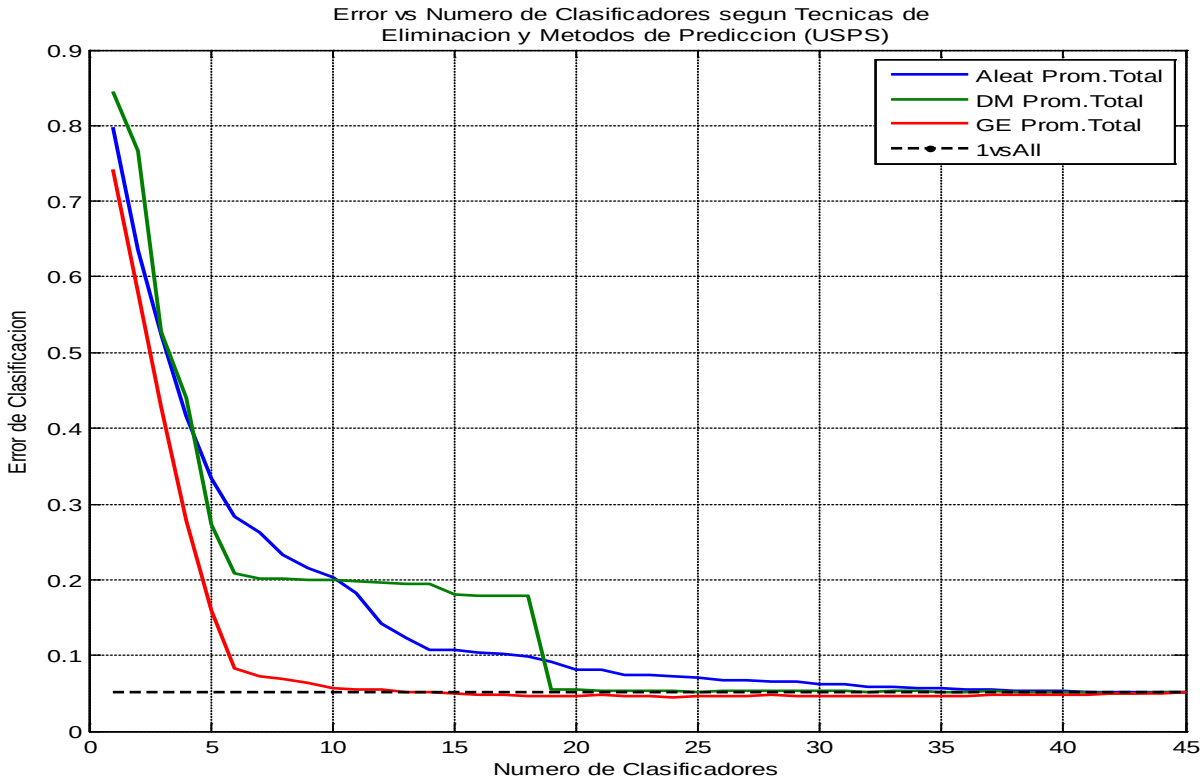


Figura 4.11: Comparación de la evolución del error de clasificación en función del número de clasificadores de las tres técnicas de eliminación y la mejor estrategia de predicción para cada una para USPS

A continuación se muestran resultados numéricos del método de eliminación de camino "greedy" de mínimo error con la estrategia de combinación por *Promedio Total* ya que se ha comprobado que es aquel que mejores resultados da.

En las siguientes tablas se muestran, para este método y los dos conjuntos de datos, en primer lugar, el error de clasificación en función del número de clasificadores entrenados, donde también puede verse el orden en que los clasificadores binarios han sido eliminados.

#Clasif	Clasif Elim		Error Clasif	#Clasif	Clasif Elim		Error Clasif	#Clasif	Clasif Elim		Error Clasif
45	1	2	10,20%	**30**	4	7	8,80%	**15**	7	10	9,80%
44	6	9	9,90%	**29**	9	10	8,80%	**14**	1	7	9,90%
43	1	9	9,70%	**28**	2	3	8,80%	**13**	5	10	10,10%
42	4	10	9,60%	**27**	2	7	8,90%	**12**	3	5	10,30%
41	3	9	9,50%	**26**	2	8	9,00%	**11**	1	8	10,70%
40	3	8	9,40%	**25**	6	10	9,10%	**10**	2	10	11,90%
39	6	7	9,30%	**24**	2	9	9,00%	**9**	7	8	13,20%
38	2	6	9,20%	**23**	8	10	9,10%	**8**	5	6	14,60%
37	4	6	9,20%	**22**	5	7	9,10%	**7**	5	9	15,70%
36	7	9	9,20%	**21**	1	6	9,10%	**6**	1	5	22,10%
35	5	8	9,10%	**20**	4	5	9,20%	**5**	3	7	28,80%
34	1	3	9,00%	**19**	8	9	9,20%	**4**	6	8	36,00%
33	1	10	8,90%	**18**	1	4	9,30%	**3**	4	8	44,10%
32	2	4	8,80%	**17**	4	9	9,60%	**2**	2	5	61,80%
31	3	4	8,80%	**16**	3	6	9,70%	**1**	3	10	80,30%

Tabla 4.11: Error de clasificación para la técnica de eliminación por camino "greedy" de mínimo error para la estrategia de predicción *Promedio Total* para *10Newsgroups*

#Clasif	Clasif Elim		Error Clasif	#Clasif	Clasif Elim		Error Clasif	#Clasif	Clasif Elim		Error Clasif
45	1	3	5,08%	**30**	6	9	4,68%	**15**	8	10	4,93%
44	2	3	5,03%	**29**	1	2	4,68%	**14**	4	10	5,18%
43	1	10	4,98%	**28**	2	7	4,73%	**13**	1	6	5,23%
42	5	6	4,93%	**27**	2	4	4,58%	**12**	3	6	5,43%
41	6	10	4,88%	**26**	9	10	4,58%	**11**	3	5	5,53%
40	6	8	4,83%	**25**	5	9	4,58%	**10**	7	9	5,68%
39	2	6	4,73%	**24**	1	7	4,53%	**9**	3	9	6,33%
38	3	4	4,73%	**23**	7	8	4,58%	**8**	2	9	6,83%
37	4	8	4,73%	**22**	2	8	4,63%	**7**	5	8	7,32%
36	4	6	4,68%	**21**	3	8	4,73%	**6**	6	7	8,32%
35	5	7	4,68%	**20**	7	10	4,63%	**5**	1	8	15,99%
34	4	9	4,68%	**19**	5	10	4,63%	**4**	4	7	27,85%
33	4	5	4,68%	**18**	1	9	4,68%	**3**	8	9	42,75%
32	3	7	4,68%	**17**	1	4	4,78%	**2**	2	5	58,05%
31	2	10	4,68%	**16**	1	5	4,83%	**1**	3	10	74,24%

Tabla 4.12: Error de clasificación para la técnica de eliminación por camino “greedy” de mínimo error para la estrategia de predicción *Promedio Total* para *USPS*

Vamos a comparar los errores obtenidos presentados en las tablas anteriores con la estrategia *1-vs-All* en el que se conseguía, para sólo 10 clasificadores, un error de 9.70% y de 5.08% para *10Newsgroups* y *USPS*, respectivamente. Para el primer conjunto, el mínimo error obtenido es del 8.80% para 28 clasificadores (casi un 1% menos), pero podemos conseguir el mismo error que en *1-vs-All* para 16 y para 10 clasificadores el error sólo se ha incrementado en un 2.2%. En el caso del conjunto *USPS*, el mínimo error se produce para 24 clasificadores y es del 4.53% y se iguala para unos 14 ó 15. En esta ocasión, podríamos reducir incluso más el número de clasificadores hasta unos 8 ó 7 ya que el error sólo se incrementa en un 1.75% y un 2.3%, respectivamente. A la vista de estos resultados, podríamos decir que sería posible reducir el número de clasificadores que debemos entrenar sin reducir considerablemente su eficiencia pero sí su coste computacional y temporal.

En las siguientes tablas se muestran las medidas F_1 para ambos conjuntos:

#Clasif	Grp1	Grp2	Grp3	Grp4	Grp5	Grp6	Grp7	Grp8	Grp9	Grp10
1-vs-All (simulación)	90,10%	93,07%	93,47%	94,00%	84,06%	90,29%	87,63%	88,54%	87,63%	94,12%
US-MSVM (artículo)	87,5%	95,4%	81,2%	90,7%	85,0%	88,5%	73,0%	93,3%	80,2%	87,2%
Pairwise (simulación)	87,44%	90,29%	93,33%	92,54%	85,58%	89,76%	90,00%	89,01%	85,13%	95,00%
45										
44	88,56%	91,18%	93,33%	92,54%	86,41%	89,76%	90,55%	89,01%	84,69%	95,00%
43	88,56%	90,73%	93,88%	92,54%	86,41%	90,73%	90,55%	89,01%	85,57%	95,00%
42	87,68%	90,73%	93,88%	92,54%	87,38%	90,73%	90,55%	89,01%	86,46%	95,00%
41	87,68%	90,73%	93,88%	92,54%	87,38%	90,73%	91,00%	89,01%	86,46%	95,52%
40	87,68%	90,73%	93,88%	92,54%	87,92%	90,73%	91,00%	89,01%	86,91%	95,52%
39	87,68%	90,73%	94,42%	92,54%	88,46%	90,29%	91,46%	88,89%	86,91%	95,52%
38	87,25%	90,29%	94,95%	92,54%	88,46%	91,18%	90,91%	89,36%	86,91%	96,04%
37	87,25%	90,73%	94,95%	92,54%	88,04%	90,64%	91,46%	89,36%	86,91%	96,04%
36	87,25%	91,18%	94,95%	93,07%	88,04%	89,66%	91,92%	89,36%	86,46%	96,04%
35	87,25%	91,18%	94,95%	93,07%	88,04%	90,64%	91,46%	89,36%	87,37%	95,57%
34	87,25%	91,18%	94,47%	93,07%	87,20%	91,54%	91,46%	89,95%	88,30%	95,57%

33	88,00%	90,73%	94,95%	93,07%	86,92%	91,54%	91,46%	90,53%	88,30%	95,57%
32	89,00%	91,18%	94,95%	93,07%	86,51%	91,54%	91,46%	90,53%	88,30%	95,57%
31	89,00%	91,18%	94,95%	93,07%	86,51%	91,54%	91,46%	90,53%	88,30%	95,57%
30	89,00%	91,18%	94,95%	93,07%	86,51%	92,00%	91,46%	90,53%	87,83%	95,57%
29	89,00%	91,18%	94,42%	93,14%	86,51%	92,00%	92,39%	90,05%	87,83%	95,57%
28	89,00%	91,18%	94,42%	93,14%	86,51%	91,54%	92,39%	90,05%	88,30%	95,57%
27	89,00%	92,08%	93,88%	92,68%	86,11%	91,54%	92,39%	89,58%	88,30%	95,57%
26	89,00%	91,18%	93,88%	93,14%	86,11%	90,64%	92,31%	90,05%	88,30%	95,57%
25	89,00%	90,73%	93,88%	93,14%	85,98%	90,20%	92,31%	90,05%	88,30%	95,57%
24	89,00%	90,73%	93,88%	93,14%	85,98%	91,54%	91,84%	90,63%	87,83%	95,57%
23	89,45%	90,57%	93,88%	93,14%	85,98%	91,92%	91,84%	90,05%	86,63%	95,57%
22	89,45%	90,57%	93,88%	92,23%	86,51%	91,92%	91,37%	91,58%	86,63%	95,00%
21	89,00%	90,57%	93,88%	92,23%	86,38%	91,92%	91,84%	91,58%	86,63%	95,05%
20	88,44%	90,48%	93,88%	92,61%	86,92%	90,10%	92,39%	91,58%	86,63%	95,05%
19	88,44%	90,05%	93,88%	93,47%	86,79%	90,20%	92,39%	91,58%	86,17%	95,10%
18	88,44%	91,79%	93,88%	93,47%	86,79%	89,66%	92,39%	90,16%	85,26%	95,10%
17	88,44%	91,79%	93,40%	91,92%	86,79%	89,66%	91,92%	90,16%	85,26%	94,58%
16	88,89%	92,68%	93,88%	90,64%	86,38%	89,22%	91,37%	90,16%	85,11%	94,58%
15	88,44%	93,66%	92,86%	92,54%	86,11%	87,38%	89,69%	90,72%	86,17%	94,53%
14	88,44%	93,20%	92,86%	92,54%	86,64%	87,80%	88,66%	90,72%	86,17%	94,00%
13	90,26%	93,20%	92,86%	92,08%	85,32%	87,38%	87,18%	90,72%	86,17%	94,00%
12	89,80%	93,20%	92,86%	91,63%	86,12%	87,08%	87,18%	89,34%	86,17%	93,53%
11	89,80%	93,20%	90,55%	92,08%	85,99%	87,08%	87,18%	87,31%	85,56%	94,00%
10	88,30%	92,75%	90,10%	91,63%	83,00%	86,67%	86,60%	82,46%	85,41%	94,00%
9	87,70%	89,52%	90,00%	91,18%	83,25%	86,26%	84,82%	81,13%	83,98%	89,86%
8	88,17%	88,46%	89,45%	91,63%	83,16%	85,17%	75,47%	81,52%	81,56%	89,66%
7	86,34%	87,62%	90,45%	91,09%	81,32%	78,76%	76,78%	81,13%	80,00%	90,00%
6	85,41%	84,79%	89,11%	88,89%	72,48%	77,73%	73,97%	78,38%	0,00%	89,55%
5	0,00%	83,64%	85,71%	88,89%	60,61%	77,39%	75,35%	79,64%	0,00%	90,00%
4	0,00%	76,27%	68,06%	88,04%	60,26%	77,59%	0,00%	78,38%	0,00%	87,20%
3	0,00%	72,58%	59,57%	82,46%	59,49%	0,00%	0,00%	70,50%	0,00%	83,49%
2	0,00%	60,26%	42,15%	0,00%	49,87%	0,00%	0,00%	0,00%	0,00%	78,69%
1	0,00%	0,00%	24,44%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	50,26%

Tabla 4.13: Medida F_1 para la técnica de eliminación por camino "greedy" de mínimo error para la estrategia de predicción *Promedio Total* para *10Newsgroups*. Comparación con *1-vs-All* y *US-MSVM*

#Clasif	**"0"**	**"1"**	**"2"**	**"3"**	**"4"**	**"5"**	**"6"**	**"7"**	**"8"**	**"9"**
1-vs-All (simulación)	97,25%	97,69%	92,27%	93,54%	92,38%	92,35%	95,52%	95,83%	93,62%	95,48%
US-MSVM (artículo)	93,5%	95,8%	88,0%	89,1%	93,1%	86,9%	95,5%	91,7%	90,3%	94,2%
Pairwise (simulación)	98,20%	97,89%	91,41%	93,29%	93,33%	92,64%	94,40%	96,53%	92,73%	95,56%
45	98,06%	97,50%	91,04%	93,87%	92,57%	92,97%	94,96%	96,53%	93,37%	95,26%
44	98,06%	97,50%	91,32%	94,15%	92,57%	92,97%	94,96%	96,53%	93,37%	95,26%
43	98,19%	97,50%	91,58%	94,15%	92,57%	92,97%	94,96%	96,53%	93,37%	95,26%
42	98,47%	97,50%	91,13%	94,15%	92,84%	93,25%	94,96%	96,53%	93,37%	95,26%
41	98,46%	97,50%	91,13%	94,15%	92,84%	93,83%	95,29%	96,53%	93,09%	95,26%
40	98,60%	97,50%	91,13%	94,15%	92,84%	93,83%	95,86%	96,53%	93,09%	95,56%
39	98,60%	97,50%	91,58%	94,15%	92,84%	93,25%	95,86%	96,53%	93,09%	95,56%
38	98,60%	97,50%	91,58%	94,15%	92,84%	93,25%	95,86%	96,53%	93,09%	95,56%
37	98,60%	97,50%	91,81%	94,15%	92,84%	93,25%	96,14%	96,53%	92,81%	95,84%
36	98,60%	97,50%	91,85%	94,15%	92,80%	93,25%	95,58%	96,53%	93,37%	95,84%
35	98,60%	97,50%	91,63%	94,15%	92,80%	93,54%	95,58%	96,53%	93,37%	95,84%
34	98,60%	97,68%	91,63%	94,15%	93,00%	93,54%	95,58%	96,53%	93,09%	95,60%
33	98,60%	97,88%	91,40%	93,87%	92,77%	93,50%	95,58%	96,53%	93,66%	95,60%
32	98,60%	97,88%	91,13%	93,87%	92,77%	93,83%	95,58%	96,53%	93,66%	95,60%
31	98,60%	97,88%	91,00%	94,44%	92,54%	94,12%	95,29%	96,53%	93,66%	95,60%
30	98,60%	97,69%	91,00%	94,44%	92,54%	94,12%	95,29%	96,53%	93,66%	95,87%
29	98,60%	97,69%	91,32%	94,15%	92,54%	92,97%	95,29%	96,89%	93,66%	95,87%
28	98,74%	97,69%	92,46%	93,87%	92,57%	92,97%	95,29%	96,89%	93,98%	95,87%

27	98,60%	97,69%	91,92%	93,58%	92,68%	93,25%	96,14%	96,53%	93,98%	96,13%
26	98,60%	97,69%	92,19%	93,58%	92,68%	93,25%	96,14%	96,53%	93,98%	95,84%
25	98,74%	97,69%	92,19%	93,29%	92,68%	93,54%	96,14%	96,53%	93,98%	96,13%
24	98,74%	97,50%	92,19%	93,29%	92,68%	93,54%	96,14%	96,19%	93,98%	96,13%
23	98,74%	97,50%	91,92%	93,29%	92,68%	93,54%	96,14%	95,86%	93,98%	96,13%
22	98,46%	97,50%	91,23%	93,58%	92,68%	93,54%	96,14%	96,19%	93,66%	96,13%
21	98,46%	97,50%	91,92%	93,21%	92,91%	94,15%	96,14%	96,19%	93,77%	95,87%
20	98,60%	97,50%	92,19%	93,21%	92,91%	94,15%	96,14%	95,83%	93,77%	95,60%
19	98,60%	97,50%	92,19%	92,92%	92,91%	93,83%	96,14%	95,83%	93,77%	95,60%
18	98,60%	97,68%	92,19%	93,25%	91,67%	93,83%	96,14%	96,53%	93,77%	94,79%
17	98,60%	97,49%	91,73%	93,25%	91,63%	93,25%	96,14%	96,53%	93,49%	95,87%
16	98,32%	97,29%	91,73%	93,25%	91,67%	93,21%	96,14%	96,19%	93,49%	95,87%
15	97,78%	97,29%	91,69%	93,87%	91,67%	93,21%	95,21%	96,19%	92,31%	95,58%
14	97,78%	97,49%	92,62%	94,15%	89,32%	93,79%	95,21%	96,19%	92,58%	95,34%
13	97,78%	96,69%	92,86%	94,15%	89,59%	93,79%	93,77%	95,86%	92,58%	95,34%
12	97,78%	96,89%	92,86%	93,50%	89,37%	93,75%	93,77%	95,86%	92,26%	95,08%
11	97,78%	96,89%	92,89%	92,97%	89,59%	92,50%	94,33%	95,86%	91,29%	95,08%
10	97,78%	96,48%	92,89%	92,97%	87,68%	92,50%	94,33%	94,41%	91,57%	91,58%
9	97,78%	96,08%	92,62%	88,89%	87,68%	92,11%	94,33%	93,99%	90,91%	92,02%
8	97,78%	95,67%	89,83%	88,63%	87,02%	92,11%	93,77%	92,09%	91,41%	92,86%
7	97,76%	95,67%	88,32%	86,49%	90,59%	89,91%	93,37%	92,42%	81,60%	93,63%
6	91,72%	96,08%	88,32%	86,14%	89,22%	82,82%	0,00%	92,42%	67,25%	93,63%
5	90,09%	0,00%	85,85%	84,62%	60,27%	83,80%	0,00%	92,42%	65,81%	93,63%
4	88,97%	0,00%	64,06%	64,00%	0,00%	81,52%	0,00%	0,00%	59,81%	61,93%
3	80,09%	0,00%	52,27%	36,78%	0,00%	62,68%	0,00%	0,00%	0,00%	0,00%
2	61,54%	0,00%	0,00%	23,42%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
1	98,06%	97,50%	91,04%	93,58%	92,57%	92,64%	94,96%	96,53%	93,37%	95,26%

Tabla 4.14: Medida F_1 para la técnica de eliminación por camino "greedy" de mínimo error para la estrategia de predicción *Promedio Total* para *USPS*. Comparación con *1-vs-All* y *US-MSVM*

4.5.5. Métodos Constructivos

En esta parte del trabajo, analizaremos varios métodos de construcción equivalentes a los métodos deconstructivos realizados anteriormente descritos en la sección 3.2.2 de este proyecto. Como ya se ha explicado es necesario hacer este tipo de métodos ya que reducen la complejidad y los costes temporales y computacionales debido a que no es necesario entrenar todos los clasificadores binarios SVM sino solamente los realmente necesarios. Se van a realizar los diversos experimentos de construcción utilizando las estrategias de combinación o predicción utilizadas en los demás experimentos.

4.5.5.1. Método "baseline": Construcción por adición de clasificadores pareados de manera Aleatoria

Esta técnica es totalmente equivalente a la de eliminación aleatoria por lo que los resultados obtenidos bajo este supuesto son similares a los logrados por el método deconstructivo. Por este motivo, no se realizarán las pertinentes simulaciones ya que ambos tienen en mismo comportamiento estadístico.

4.5.5.2. Construcción por adición de clasificadores pareados basada en Distancias Mínimas

Esta técnica esta basada en el método US-MSVM propuesto por los autores en el artículo en la que se han eliminado las restricciones de distancia umbral y número máximo de clasificadores.

Se entrenan, progresivamente, un número cada vez mayor de clasificadores SVM hasta entrenar el número máximo de clasificadores posibles para el número de clases del problema, en nuestro caso 45. Por tanto, en la primera iteración del método, se entrena únicamente 1 clasificador, en la segunda 2, y así sucesivamente hasta haber entrenado los 45 clasificadores en la última iteración del bucle. En cada una de dichas iteraciones, se parte de un par de clases elegidas aleatoriamente para entrenar el primer clasificador y nos basamos en la estrategia de muestreo de incertidumbre para elegir los próximos clasificadores necesarios en esa iteración.

De este modo, se estudia la influencia del número de clasificadores que hay que entrenar en los resultados finales como en el error de clasificación de test. Esta evolución, para los dos conjuntos de datos *10Newsgroups* y *USPS*, puede verse gráficamente en la siguiente figura.

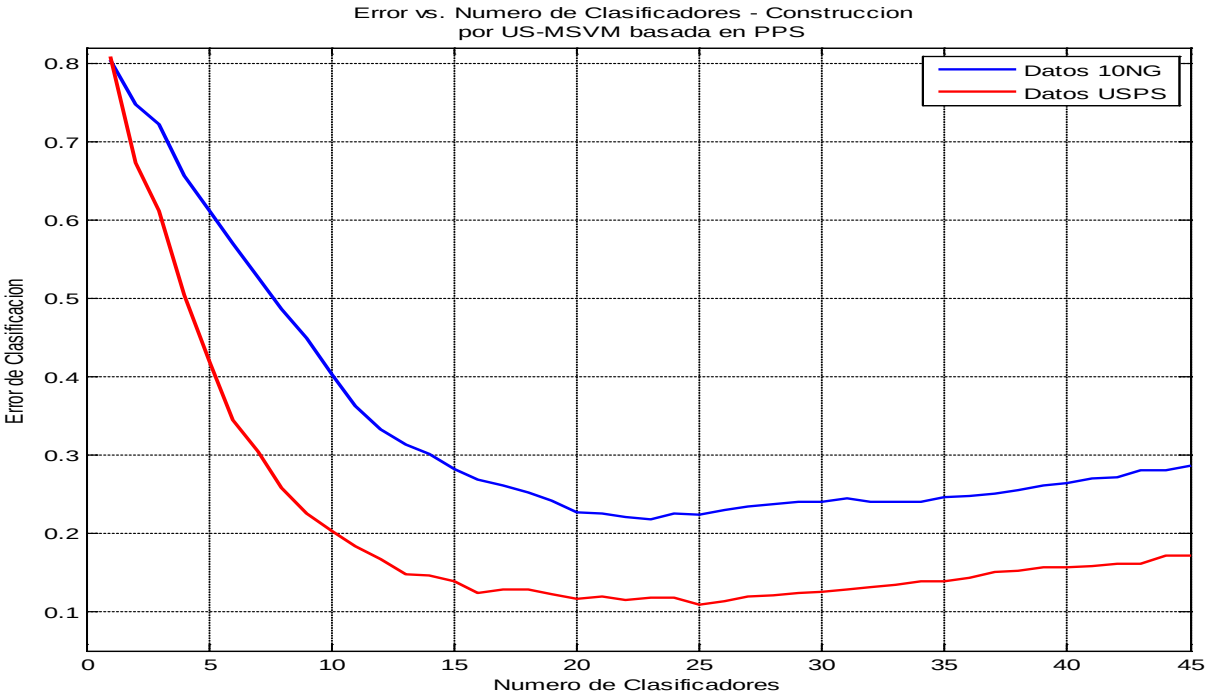


Figura 4.12: Evolución del error de clasificación según el número de clasificadores para el método de construcción basado en US-MSVM para ambas colecciones de datos

Puede comprobarse gráficamente el comportamiento del método US-MSVM que nos habían presentado los autores el en artículo. En éste se comenta que en la técnica propuesta se entrenan los clasificadores por orden de importancia siendo los primeros en entrenarse los más “útiles” y que existe un cierto umbral en el que se empeora el rendimiento del método ya que se empiezan a entrenar clasificadores que no se consideran muy “útiles”. En la Figura 2.11 que se ha presentado en la sección 2.3.3 del presente proyecto, puede verse

la influencia del número de clasificadores entrenados en los resultados de precisión y exhaustividad que han sido publicados en el artículo. Se comprueba que a medida que aumenta el número de clasificadores se mejoran los resultados para ambas colecciones de datos y que a partir de un cierto número no se producen mejoras muy significativas o incluso empeoran las prestaciones del método.

En la Figura 4.12 se presenta la evolución del error según el número de clasificadores para nuestro experimento basado en el método US-MSVM. Se puede ver que el umbral donde comienza a empeorar el rendimiento para el conjunto *10Newsgroups* se encuentra en unos 23 clasificadores donde justo se consigue el error de clasificación mínimo, 21.82%, y para *USPS* está entorno a los 25 clasificadores donde se logra también el error mínimo del 10.87%.

A continuación se muestran las tablas del error de clasificación, donde también se puede ver en que orden en que han sido entrenados los 45 clasificadores para ambos conjuntos de prueba:

#Clasif	Clasif Entrenado		Error Clasif	#Clasif	Clasif Entrenado		Error Clasif	#Clasif	Clasif Entrenado		Error Clasif
1	2	10	80,49%	**16**	5	8	26,79%	**31**	4	5	24,52%
2	2	5	74,83%	**17**	1	9	26,17%	**32**	2	3	24,00%
3	3	7	72,21%	**18**	4	7	25,23%	**33**	4	10	23,97%
4	4	6	65,71%	**19**	6	7	24,12%	**34**	6	9	24,05%
5	8	9	61,19%	**20**	2	6	22,73%	**35**	1	2	24,67%
6	1	5	56,99%	**21**	4	8	22,55%	**36**	8	10	24,77%
7	7	10	52,69%	**22**	5	6	22,16%	**37**	1	7	25,00%
8	5	9	48,59%	**23**	3	4	21,82%	**38**	5	7	25,45%
9	3	6	44,85%	**24**	1	8	22,49%	**39**	1	3	26,10%
10	6	8	40,34%	**25**	2	7	22,38%	**40**	3	5	26,37%
11	2	8	36,25%	**26**	4	9	23,01%	**41**	7	9	27,04%
12	1	4	33,31%	**27**	3	8	23,50%	**42**	3	9	27,12%
13	7	8	31,33%	**28**	6	10	23,75%	**43**	5	10	28,02%
14	3	10	30,13%	**29**	1	6	24,05%	**44**	1	10	28,00%
15	2	4	28,21%	**30**	2	9	24,00%	**45**	9	10	28,70%

Tabla 4.15: Error de clasificación para la técnica de construcción basada en distancias máximas (estrategia de US-MSVM) para *10Newsgroups*

#Clasif	Clasif Entrenado		Error Clasif	#Clasif	Clasif Entrenado		Error Clasif	#Clasif	Clasif Entrenado		Error Clasif
1	7	9	80,93%	**16**	2	7	12,45%	**31**	3	4	12,83%
2	1	4	67,30%	**17**	2	10	12,82%	**32**	3	7	13,14%
3	5	6	61,12%	**18**	2	3	12,82%	**33**	3	6	13,50%
4	3	5	50,38%	**19**	2	6	12,21%	**34**	3	9	13,81%
5	4	10	41,96%	**20**	4	7	11,64%	**35**	1	7	13,82%
6	5	10	34,47%	**21**	3	10	11,98%	**36**	5	7	14,36%
7	8	9	30,46%	**22**	1	8	11,56%	**37**	5	9	15,15%
8	2	8	25,80%	**23**	7	8	11,76%	**38**	1	6	15,15%
9	3	8	22,59%	**24**	4	6	11,82%	**39**	7	10	15,60%
10	4	9	20,36%	**25**	4	5	10,87%	**40**	1	5	15,65%
11	2	4	18,41%	**26**	1	9	11,32%	**41**	6	7	15,82%
12	2	5	16,72%	**27**	6	8	11,88%	**42**	6	9	16,10%
13	6	10	14,72%	**28**	2	9	12,13%	**43**	1	3	16,14%
14	5	8	14,64%	**29**	8	10	12,39%	**44**	9	10	17,09%
15	4	8	13,94%	**30**	1	2	12,51%	**45**	1	10	17,14%

Tabla 4.16: Error de clasificación para la técnica de construcción basada en distancias máximas (estrategia de US-MSVM) para *USPS*

A continuación se presentan las tablas de la medida F_1 para esta estrategia de construcción y para ambos conjuntos:

#Clasif	Grp1	Grp2	Grp3	Grp4	Grp5	Grp6	Grp7	Grp8	Grp9	Grp10
1	15,12%	2,81%	8,58%	5,69%	3,56%	5,81%	6,42%	3,19%	6,97%	9,93%
2	34,29%	11,97%	6,17%	3,80%	11,92%	0,00%	6,54%	7,31%	9,22%	39,08%
3	35,28%	15,73%	28,18%	20,51%	17,67%	0,00%	10,23%	6,39%	13,15%	42,32%
4	38,54%	18,87%	30,91%	30,20%	25,49%	8,70%	18,35%	15,47%	36,53%	57,16%
5	47,70%	27,96%	40,66%	29,00%	22,13%	20,89%	29,53%	28,26%	43,26%	49,31%
6	49,17%	24,63%	41,41%	40,18%	33,85%	28,20%	36,02%	24,30%	48,89%	67,84%
7	54,19%	40,12%	49,46%	35,33%	33,06%	32,81%	44,56%	34,37%	46,48%	78,59%
8	57,31%	36,76%	49,12%	42,79%	48,31%	41,69%	46,71%	31,50%	52,27%	83,98%
9	59,41%	41,33%	62,30%	40,81%	52,58%	45,87%	46,74%	44,14%	60,97%	82,54%
10	66,71%	44,69%	66,42%	47,59%	63,50%	44,01%	57,48%	45,05%	60,56%	87,13%
11	71,37%	55,70%	67,46%	49,27%	65,83%	57,00%	60,35%	51,36%	64,88%	86,11%
12	68,27%	59,39%	72,40%	53,41%	66,27%	64,28%	65,96%	58,89%	66,39%	87,59%
13	67,21%	57,29%	75,85%	57,27%	71,48%	70,18%	67,42%	62,60%	67,07%	87,34%
14	70,95%	59,46%	75,29%	57,03%	66,35%	65,23%	71,01%	69,00%	70,68%	90,11%
15	70,22%	63,26%	79,04%	57,69%	73,37%	69,42%	72,35%	69,47%	67,27%	91,70%
16	72,65%	67,04%	78,45%	64,20%	70,93%	68,32%	74,21%	70,51%	71,76%	92,16%
17	75,44%	65,88%	77,48%	61,50%	75,94%	69,68%	72,13%	73,17%	72,47%	92,16%
18	74,56%	71,97%	78,31%	59,54%	76,91%	71,81%	75,15%	73,20%	71,00%	92,15%
19	76,27%	74,96%	77,00%	64,32%	76,82%	73,04%	75,61%	73,92%	72,21%	92,52%
20	76,93%	76,18%	76,82%	67,85%	77,06%	75,76%	80,13%	75,35%	73,00%	92,24%
21	76,70%	74,77%	79,04%	69,74%	78,34%	76,93%	78,48%	74,73%	72,90%	92,23%
22	75,25%	75,58%	77,81%	74,02%	76,68%	77,47%	79,05%	76,98%	73,03%	92,29%
23	76,87%	75,39%	80,28%	76,60%	76,69%	76,66%	80,58%	74,90%	71,96%	91,63%
24	76,27%	74,20%	78,64%	75,79%	77,43%	76,00%	78,33%	75,85%	71,48%	91,00%
25	76,62%	73,94%	79,50%	74,95%	78,42%	74,33%	78,62%	76,95%	71,40%	91,21%
26	76,56%	74,69%	76,77%	73,96%	78,06%	74,57%	75,66%	75,83%	71,75%	91,71%
27	75,76%	75,15%	75,88%	72,82%	76,76%	74,92%	72,18%	77,07%	72,36%	91,97%
28	75,56%	76,11%	74,80%	72,91%	77,43%	72,98%	71,63%	76,52%	71,88%	92,25%
29	74,30%	76,49%	75,58%	74,81%	76,34%	72,42%	70,24%	74,56%	72,32%	91,69%
30	75,41%	75,99%	75,95%	73,52%	75,83%	74,35%	70,48%	73,14%	72,87%	92,10%
31	74,24%	77,06%	74,97%	74,42%	73,90%	73,39%	69,38%	72,96%	71,96%	91,96%
32	74,65%	78,49%	75,96%	77,32%	73,37%	72,54%	69,86%	73,29%	70,97%	92,23%
33	74,26%	78,50%	76,44%	78,21%	71,97%	72,25%	70,85%	71,78%	72,17%	92,48%
34	75,01%	78,78%	75,33%	76,88%	71,50%	73,13%	70,93%	71,93%	72,96%	91,88%
35	74,20%	77,94%	74,80%	77,61%	70,43%	72,24%	67,60%	73,36%	72,76%	90,62%
36	74,14%	77,52%	74,15%	77,28%	70,83%	72,42%	67,81%	73,07%	72,86%	90,37%
37	75,00%	77,13%	75,12%	76,47%	70,10%	71,74%	66,67%	73,63%	72,25%	89,86%
38	72,91%	77,35%	74,38%	75,07%	71,21%	71,37%	67,73%	71,64%	72,22%	89,86%
39	72,57%	77,73%	73,76%	73,83%	70,28%	70,17%	67,41%	71,28%	70,87%	89,33%
40	71,22%	77,66%	73,79%	73,97%	70,73%	68,36%	67,48%	69,39%	72,87%	88,88%
41	72,16%	77,36%	73,10%	72,31%	68,68%	68,28%	67,26%	67,55%	71,51%	89,29%
42	72,78%	78,76%	72,65%	70,42%	68,34%	68,13%	67,47%	66,47%	71,74%	89,79%
43	72,89%	79,08%	70,12%	67,79%	69,18%	65,58%	65,89%	65,06%	71,97%	89,23%
44	72,89%	79,43%	70,00%	68,09%	68,69%	65,22%	65,63%	65,48%	72,46%	89,00%
45	72,89%	76,92%	69,72%	68,06%	68,69%	64,44%	64,25%	63,16%	72,46%	89,00%

Tabla 4.17: Medida F_1 para la técnica de construcción basada en distancias máximas (estrategia de US-MSVM) para *10Newsgroups*.

#Clasif	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
1	4,47%	23,90%	7,21%	6,62%	2,09%	1,80%	11,12%	14,02%	2,78%	3,54%
2	22,07%	76,94%	8,70%	14,76%	5,08%	5,80%	22,70%	12,57%	6,51%	0,00%
3	21,53%	69,97%	0,00%	17,51%	21,18%	16,56%	51,88%	44,16%	5,49%	31,21%
4	14,43%	90,48%	28,49%	45,32%	27,49%	29,08%	59,17%	62,14%	13,58%	51,51%
5	47,47%	87,95%	23,62%	61,59%	48,84%	43,14%	65,26%	71,33%	28,40%	43,28%
6	71,85%	93,68%	28,78%	67,40%	57,65%	49,09%	64,58%	77,31%	34,85%	63,15%
7	72,99%	96,48%	47,11%	69,25%	52,88%	55,59%	72,89%	82,12%	43,84%	71,38%
8	79,65%	95,58%	53,42%	81,29%	63,80%	63,38%	73,46%	80,33%	57,25%	77,38%
9	85,84%	95,38%	60,08%	86,35%	63,26%	69,10%	75,50%	79,94%	64,10%	75,64%
10	86,90%	96,30%	68,33%	87,49%	64,86%	70,72%	79,76%	78,51%	72,86%	76,72%
11	87,70%	95,76%	69,60%	87,38%	69,43%	73,36%	83,16%	79,29%	74,69%	82,55%
12	90,76%	95,91%	68,51%	89,04%	74,81%	74,02%	84,56%	82,05%	73,91%	85,52%
13	92,59%	95,46%	75,04%	87,51%	78,45%	79,33%	88,45%	83,87%	74,38%	84,93%
14	92,80%	96,15%	71,72%	87,81%	76,42%	81,98%	87,35%	86,00%	75,20%	85,79%
15	93,40%	95,65%	75,92%	89,51%	79,67%	82,08%	88,10%	82,49%	78,06%	83,69%
16	94,79%	95,91%	77,60%	87,46%	75,79%	87,78%	90,18%	88,20%	82,79%	85,14%
17	94,28%	95,73%	77,75%	88,62%	76,99%	86,34%	90,15%	88,50%	78,12%	85,20%
18	94,00%	94,80%	79,95%	89,90%	76,76%	86,37%	89,94%	89,01%	77,24%	84,32%
19	94,31%	95,34%	81,15%	89,30%	79,74%	85,60%	90,91%	88,98%	77,87%	85,00%
20	94,47%	95,07%	81,85%	89,73%	82,07%	85,77%	91,85%	91,11%	77,97%	85,11%
21	94,99%	95,14%	81,05%	89,52%	79,74%	86,39%	91,34%	91,72%	77,41%	83,65%
22	95,17%	96,69%	80,84%	89,94%	79,93%	85,73%	91,88%	91,81%	79,53%	83,41%
23	95,39%	95,77%	81,48%	90,10%	80,38%	85,56%	91,71%	91,54%	77,17%	83,07%
24	95,32%	95,97%	81,12%	88,32%	80,35%	86,69%	91,42%	92,10%	75,75%	84,27%
25	95,32%	96,96%	82,33%	89,83%	83,22%	85,94%	92,05%	92,20%	78,84%	85,19%
26	94,95%	96,86%	80,70%	89,93%	81,68%	86,64%	91,47%	92,23%	78,00%	84,92%
27	94,90%	97,27%	79,32%	89,26%	80,21%	85,08%	91,59%	92,52%	75,63%	84,88%
28	95,20%	97,25%	78,45%	89,48%	78,10%	85,75%	92,36%	92,10%	74,90%	84,10%
29	95,15%	97,23%	78,76%	89,25%	77,80%	86,30%	91,73%	91,44%	73,87%	83,10%
30	94,74%	97,56%	78,50%	88,86%	77,13%	86,76%	91,45%	91,50%	73,78%	83,26%
31	95,07%	97,58%	76,93%	89,89%	75,74%	86,28%	90,94%	90,64%	73,94%	82,87%
32	95,20%	97,33%	76,44%	89,97%	74,43%	86,02%	90,25%	90,68%	74,29%	81,77%
33	95,13%	97,11%	76,91%	89,69%	73,71%	85,82%	89,85%	90,63%	72,10%	81,13%
34	95,21%	97,23%	75,65%	89,99%	72,58%	85,98%	89,72%	89,62%	71,46%	80,79%
35	95,08%	97,50%	75,70%	89,23%	72,59%	86,40%	89,76%	89,61%	70,30%	81,43%
36	94,57%	97,48%	76,24%	85,80%	71,97%	88,03%	90,03%	89,36%	65,93%	81,30%
37	94,57%	97,32%	75,12%	86,13%	67,04%	87,82%	89,77%	89,29%	64,39%	80,00%
38	94,57%	97,49%	75,09%	86,05%	67,13%	87,66%	90,03%	89,25%	63,88%	80,00%
39	94,57%	97,34%	74,86%	85,59%	66,80%	85,90%	89,72%	89,57%	60,67%	79,97%
40	94,57%	97,33%	74,83%	85,55%	66,67%	85,62%	89,89%	89,68%	60,38%	79,71%
41	94,57%	97,31%	74,72%	85,17%	64,74%	86,20%	89,91%	89,75%	60,31%	79,30%
42	94,55%	97,33%	75,24%	83,24%	64,72%	84,78%	89,72%	90,04%	59,89%	79,16%
43	94,55%	97,33%	75,29%	83,00%	64,72%	84,62%	89,69%	90,07%	59,78%	79,14%
44	94,55%	97,33%	73,71%	81,16%	66,67%	82,96%	89,44%	89,12%	50,76%	79,14%
45	94,55%	97,14%	73,71%	81,16%	66,67%	82,96%	89,44%	89,12%	50,76%	78,85%

Tabla 4.18: Medida F_1 para la técnica de construcción basada en distancias máximas (estrategia de US-MSVM) para *USPS*.

4.5.5.3. Construcción por adición de clasificadores basada en la búsqueda de un camino de Mínimo Error de Clasificación

Este método de construcción es equivalente al realizar el camino de error mínimo de clasificación en la fase de test y, como ya se ha mencionado en el Capítulo 3 en la sección 3.2.2.3, se van a realizar dos técnicas para la elección del siguiente clasificador que se va a añadir: una basada en un algoritmo "greedy" de mínimo error y otra utilizando un matriz de error de clasificación con la que se elige el par de clases con mayor error acumulado en cada iteración del método.

I. Mediante un Algoritmo "Greedy"

Con este algoritmo "greedy" se calcula un camino de construcción por adición de clasificadores con el que se consigue el error mínimo de clasificación de test.

En las gráficas se muestra la evolución del error de clasificación según el número de clasificadores que han sido entrenados para varias técnicas de combinación final de clasificadores.

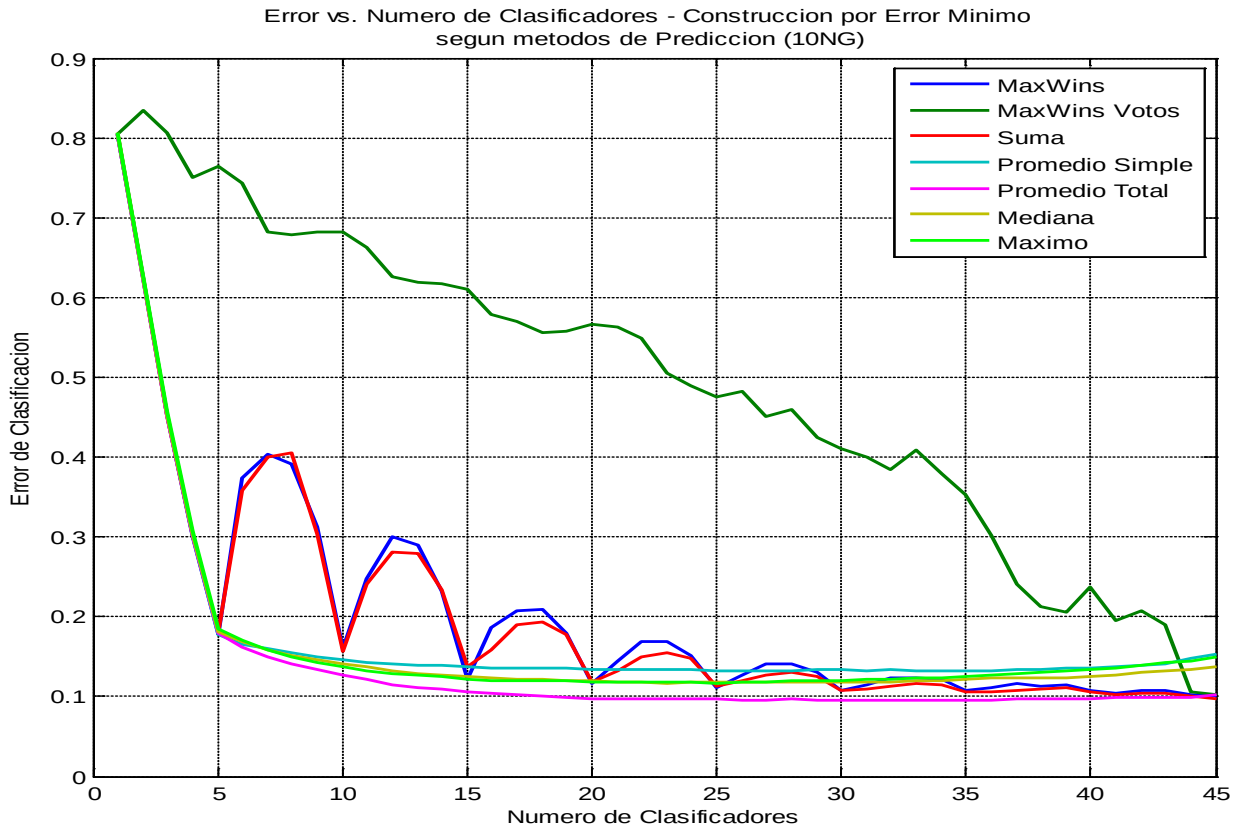


Figura 4.13: Evolución del error de clasificación según el número de clasificadores para el método de construcción basado en camino "greedy" de mínimo error y varias estrategias de predicción para *10Newsgroups*

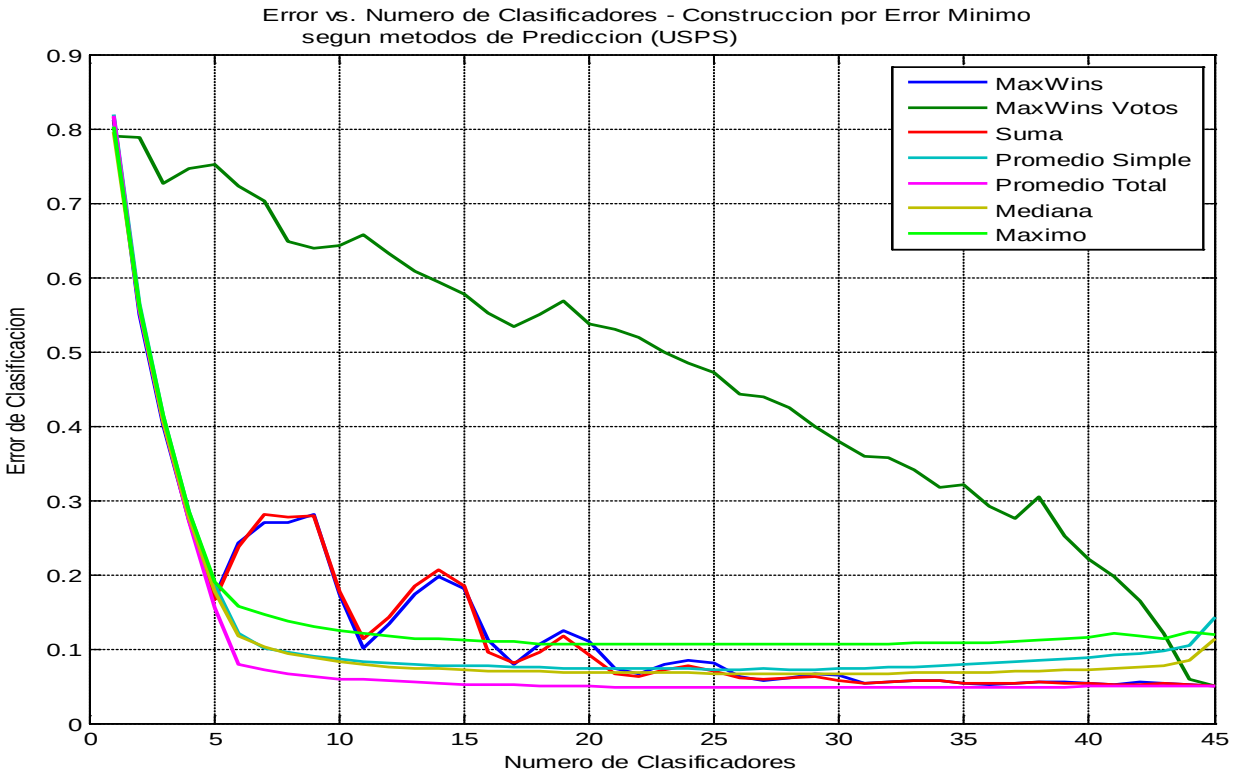


Figura 4.14: Evolución del error de clasificación según el número de clasificadores para el método de construcción basado en camino "greedy" de mínimo error y varias estrategias de predicción para *USPS*

Podemos observar que para ambos conjuntos de datos las técnicas de predicción basadas en niveles de confianzas son las que mejores resultados dan siendo el método de *Promedio Total* la que mejor comportamiento presenta para todo el número de clasificadores entrenados del problema. En cambio las técnicas basadas en voto por mayoría simple o ponderada, *MaxWins* y *MaxWins Votos*, dan malos resultados al utilizar estos dos métodos de construcción ya que no se produce una reducción progresiva del error sino múltiples variaciones según el número de clasificadores que han sido entrenados. Se puede ver claramente para este caso que el error disminuye y aumenta en bloques de unos 5 clasificadores, y este número es debido a que para las 10 clases existentes en nuestras colecciones hacen falta entrenar 5 clasificadores para tener presencia de todas ellas.

Por ejemplo, si en un principio se han entrenado los siguientes 5 clasificadores con los 5 diferentes pares formados por las 10 clases del problema:

Clasificadores Entrenados	C_1		C_2		C_3		C_4		C_5	
Par de Clases	9	10	4	6	2	3	1	7	5	8

Tabla 4.19: Ejemplo que muestra los 5 clasificadores formados por los 5 pares de clases diferentes en un problema de 10 clases

Se obtiene un error de clasificación medio del 16.70% y un error para cada clase de:

Clases	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Error (%)	17	8	14	9	32	14	13	29	14	17

Tabla 4.20: Ejemplo que muestra el error de clasificación para cada clase tras entrenarse 5 clasificadores en las que hay presencia de las 10 clases

Se elige el siguiente par de clases para entrenar el sexto clasificador, por ejemplo, el par formado por las clases 1 y 5. En este caso se aumenta el error de clasificación medio hasta un 35%. Y ahora el error de clasificación para cada clase es:

Clases	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Error (%)	8	50	33	43	4	46	23	33	80	30

Tabla 4.21: Ejemplo que muestra el error de clasificación para cada clase tras entrenarse 6 clasificadores.

Como se puede ver se ha aumentado notablemente el error de clasificación para todas las clases a excepción de aquellas, la clase 1 y la 5, que en este momento pueden aportar más votos a la clasificación de los datos. Así se aumenta el error de clasificación medio hasta un momento en el que se al haberse entrenando más clasificadores se aportan mayor número de votos a más clases y, por tanto, se reduce de nuevo el error de cada una de ellas y de este modo el error medio. Este comportamiento errante seguirá produciéndose hasta que se hayan entrenado todos los clasificadores pero más o menos se estabiliza, es decir, no se producen variaciones muy significativas del error, a partir de unos 35 clasificadores para el conjunto *10Newsgroups* y de unos 28 para *USPS*.

Como ya se ha comentado la estrategia que mejores resultados presenta es la basada en la regla *Promedio Total* y a continuación se describen dichos resultados. En el caso la colección de datos *10Newsgroups* se consigue un error de clasificación mínimo de 9.56% para 32 clasificadores y para la colección *USPS* se tiene un error de 4.75% para 25. En ambos casos podríamos reducir el número de clasificadores que tenemos que entrenar sin tener un incremento muy importante del error, en *10Newsgroups* se puede reducir hasta los 11 y en *USPS* en hasta los 7, con un incremento del error menor del 3%, con un error del 12.09% y del 7.22% respectivamente.

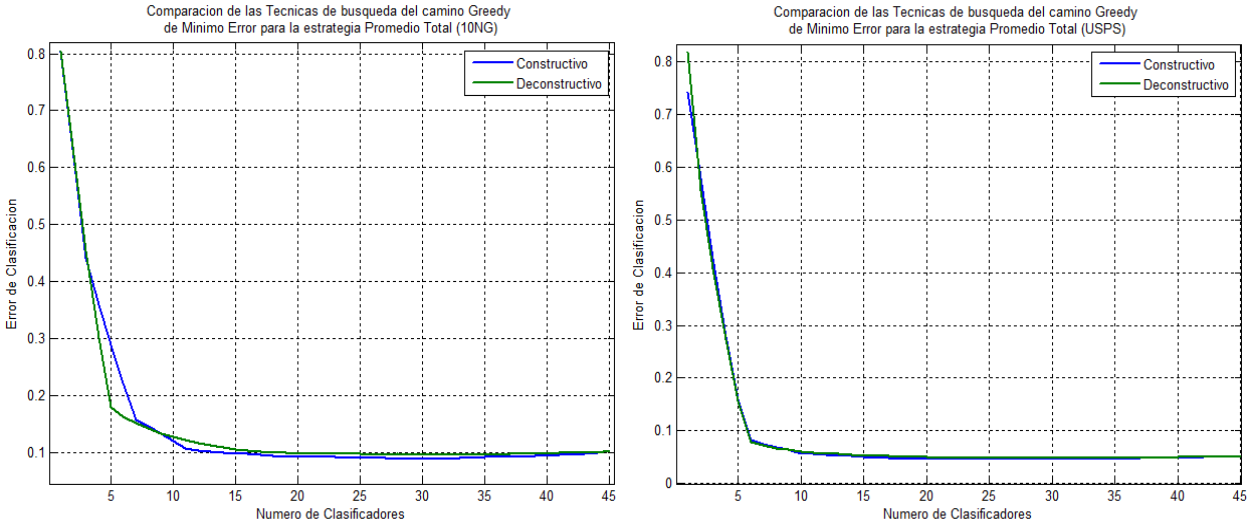


Figura 4.15: Comparación de la evolución del error de clasificación según el número de clasificadores para los métodos de construcción y deconstrucción basados en camino "greedy" de mínimo error y la estrategia *Promedio Total* para *10Newsgroups* y *USPS*

Este método de construcción, como ya habíamos mencionado, tiene un comportamiento estadísticamente similar a su versión deconstructiva. Este hecho puede verse en la Figura 4.15 donde comparamos el método constructivo y el deconstructivo basados en un

algoritmo “greedy” de búsqueda del camino de error mínimo para la estrategia *Promedio Total* y para ambos conjuntos.

Para finalizar con este método de construcción cabe comentar que no cumple el objetivo marcado de este proyecto ya que se necesita haber entrenado previamente todos los clasificadores pareados y por tanto no reducirá la carga computacional.

II. Mediante una matriz de construcción de error de clasificación

En esta técnica se utiliza una matriz de error de construcción para realizar la elección de los clasificadores a añadir para conseguir un camino de mínimo error de clasificación.

A continuación se muestra gráficamente la evolución del error de clasificación en función del número de clasificadores que han sido entrenados según diversas estrategias de predicción para ambos conjuntos de datos en estudio.

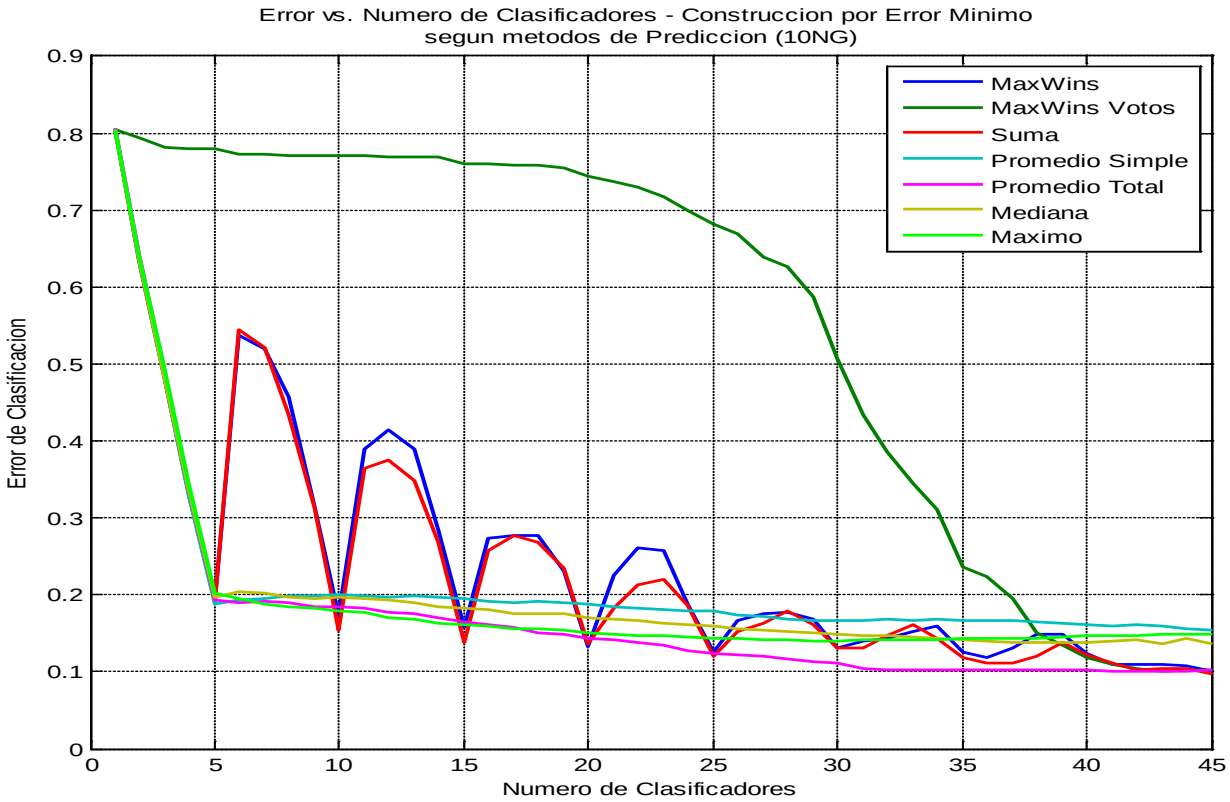


Figura 4.16**:** Evolución del error de clasificación según el número de clasificadores para el método de construcción basado en una matriz de mínimo error y varias estrategias de predicción para *10Newsgroups*

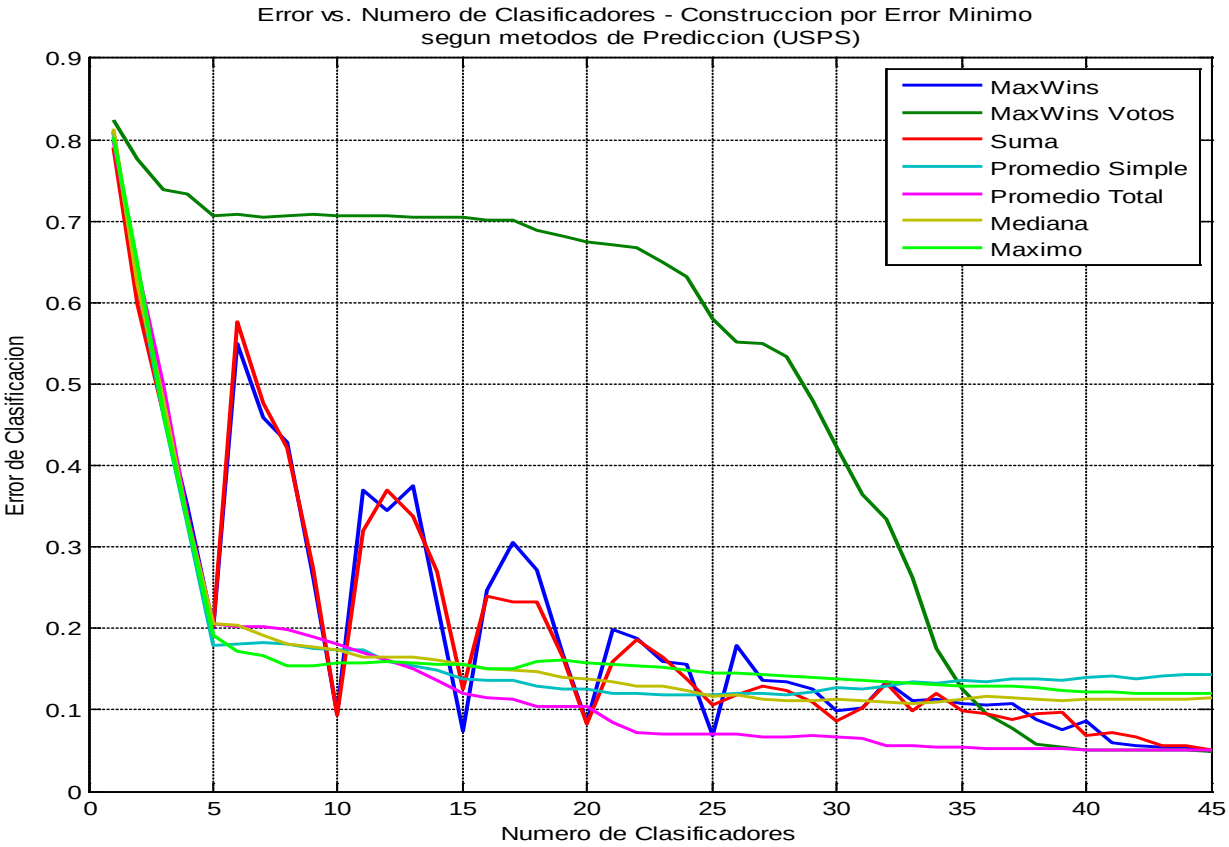


Figura 4.17: Evolución del error de clasificación según el número de clasificadores para el método de construcción basado en una matriz de mínimo error y varias estrategias de predicción para *USPS*

Como ocurría en el caso anterior con la técnica de construcción de un camino "greedy" de mínimo error, en este método y para ambos conjuntos se puede ver que las mejores técnicas de predicción son las basadas en niveles de medidas de confianza, a excepción de la basada en la regla de la *Suma*, siendo la basada en *Promedio Total* con la que mejores resultados se obtienen. También se puede observar el mal comportamiento de los métodos basados en mayoría de votos, *MaxWins* y *MaxWins Votos*, que puede explicarse de la misma manera como lo hicimos en el caso anterior.

Para esta técnica con la estrategia *Promedio Total* se han obtenido los siguientes resultados numéricos. Para ambos conjuntos de datos se obtiene el error mínimo para 42 clasificadores en *10Newsgroups* y para 43 clasificadores en *USPS*, siendo 10.07% y 5.08% respectivamente. En ambos casos podríamos reducir el número de clasificadores que tenemos que entrenar sin tener un incremento muy importante del error, menor del 3%. En *10Newsgroups* podemos reducir el número de clasificadores hasta los 24 donde se consigue un error del 12.77% y en *USPS* en hasta los 22 con un error del 7.23%.

III. Comparación de estas técnicas de construcción basadas en búsqueda de un camino de Mínimo Error de Clasificación

Para concluir, en este apartado vamos a hacer una comparación de las dos técnicas anteriores de construcción basadas en la búsqueda de un camino de mínimo error de clasificación en la fase de test.

Como se puede ver en las gráficas de la evolución del error en función del número de clasificadores para ambas técnicas de adición y ambas colecciones (Figuras 4.13 a 4.17), las técnicas de combinación que mejores resultados dan son las basadas en niveles de medidas de confianza a excepción de la *Suma*, siendo *Promedio Total* con la mejor de todas. Podemos decir que la primera de ellas basada en un algoritmo "greedy" da mejores resultados que la segunda técnica para todo el rango de clasificadores entrenados y sobretodo se mejora con un número bajo de éstos. Este hecho se puede ver en las siguientes gráficas en las que se ha utilizado la estrategia *Promedio Total* ya que es con la que mejores resultados se ha obtenido.

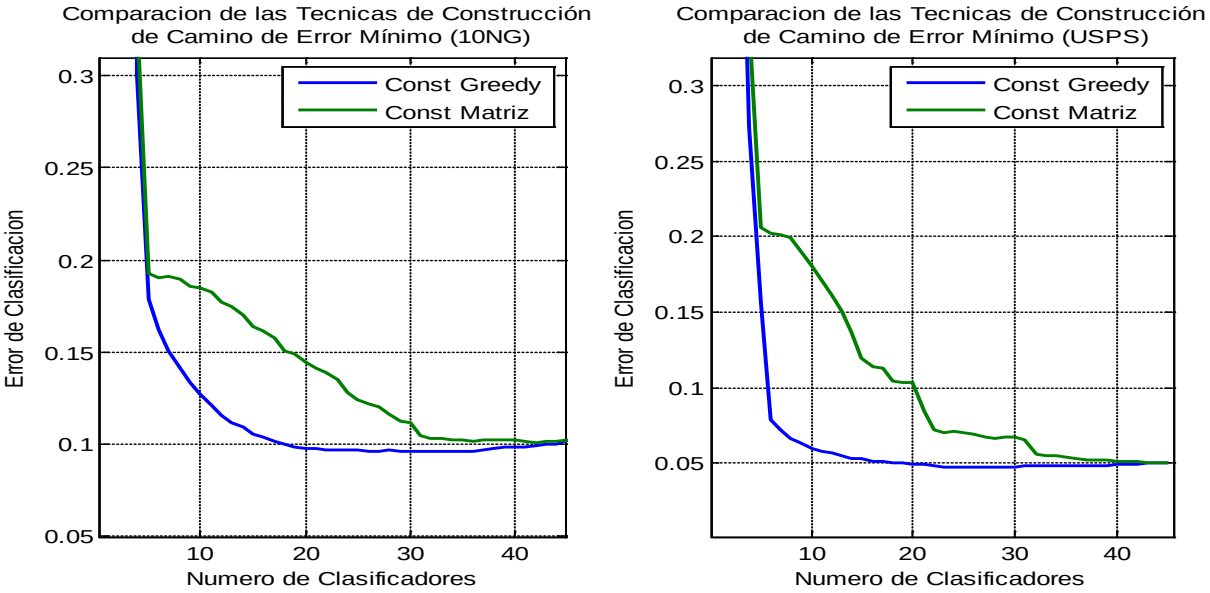


Figura 4.18: Comparación de la evolución del error de clasificación según el número de clasificadores de los métodos de búsqueda de un camino de error mínimo por un algoritmo "greedy" y por una matriz de construcción para las dos colecciones de datos *10Newsgroups* y *USPS*

En el caso de la colección *10Newsgroups* la técnica basada en la matriz de construcción empeora el error de clasificación hasta en un 6% para 13 clasificadores y en el caso de *USPS* de hasta casi un 13% para 6 clasificadores con respecto a la técnica "greedy". Para este último algoritmo en media se mejoran los resultados en más de un 2.5% para el primer conjunto y en más del 4% para el segundo.

No obstante, aunque como se acaba de comentar la técnica basada en un algoritmo "greedy" de búsqueda de un camino mínimo de error es la que mejores resultados da, tiene el principal inconveniente de que no cumple con la reducción del coste computacional ya que se ha tenido que entrenar todos los clasificadores pareados. Por este motivo, pese a la pérdida de prestaciones por el aumento del error de clasificación sobre todo para un número de clasificadores pequeño, es conveniente utilizar la técnica basada en la matriz de construcción de error mínimo. Con esta técnica podemos lograr el objetivo de este proyecto de reducir la carga computacional ya que se pueden entrenar solamente los clasificadores pareados necesarios para conseguir unos resultados de clasificación aceptables.

4.5.5.4. Comparación de las Técnicas de Construcción

Finalmente en este apartado vamos a hacer un estudio comparativo con las diferentes técnicas de construcción estudiadas, la técnica de construcción de búsqueda de un camino de error mínimo de clasificación mediante una matriz de construcción, la técnica utilizada en el artículo *US-MSVM* basada en *PPS*, y la técnica *1-vs-All*, para los dos conjuntos de prueba *10Newsgroups* y *USPS*. Se van a comparar dichas técnicas de construcción en las que se ha aplicado la estrategia de predicción o combinación *Promedio Total* ya que es con la que mejores resultados se han obtenidos.

Se muestra gráficamente la evolución del error dependiendo del número de clasificadores y las técnicas de eliminación utilizadas mostrando para cada una de ellas, únicamente, la curva de la estrategia de predicción con la se obtuvieron los mejores resultados.

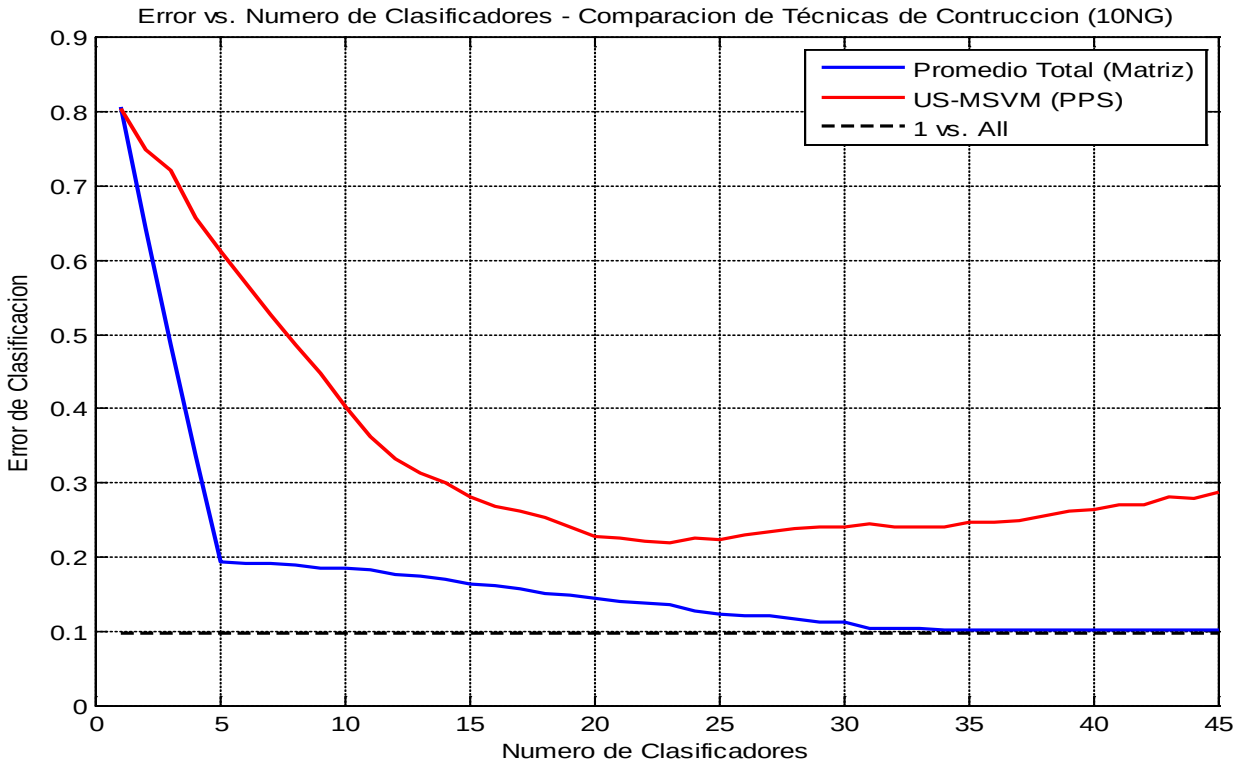


Figura 4.19: Comparación de la evolución del error de clasificación en función del número de clasificadores de las dos técnicas de construcción y varias estrategias de predicción para *10Newsgroups*

Para ambos conjuntos de datos, la técnica de construcción equivalente a la búsqueda de camino de error mínimo por matriz de construcción con la estrategia *Promedio Total* obtiene mejores resultados que con la técnica utilizada en el artículo, *US-MSVM* con la medida *PPS*. No obstante no se alcanza la cota de error marcada con *1-vs-All* para ninguna colección de prueba aunque podría reducirse el número de clasificadores sin perder demasiadas prestaciones pero si la carga temporal y computacional.

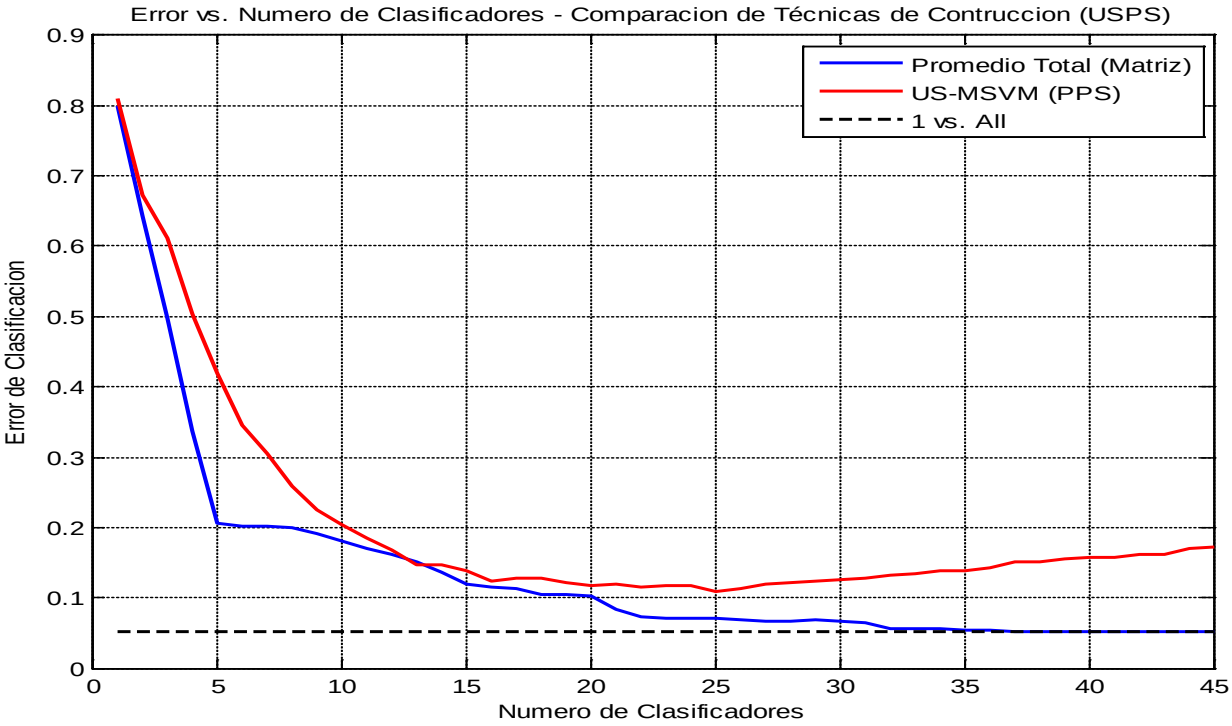


Figura 4.20: Comparación de la evolución del error de clasificación en función del número de clasificadores de las dos técnicas de construcción y varias estrategias de predicción para *USPS*

A continuación se muestran resultados numéricos del método de construcción basado en la búsqueda del camino de mínimo error basado en una matriz de construcción con la estrategia de combinación por *Promedio Total* ya que se ha comprobado que es con aquel que mejores resultados se obtiene.

En las siguientes tablas se muestran para este método y los dos conjuntos de datos el error de clasificación en la fase de test. Vamos a comparar los errores obtenidos presentados en las tablas 4.22 y 4.23 con el error de la estrategia *1-vs-All* en el que se conseguía, para sólo 10 clasificadores, un error de 9.70% para *10Newsgroups* y de 5.08% para *USPS*.

#Clasif	Error Clasif	#Clasif	Error Clasif	#Clasif	Error Clasif
1	80,61%	16	16,12%	31	10,44%
2	64,21%	17	15,75%	32	10,32%
3	48,70%	18	15,05%	33	10,32%
4	33,60%	19	14,89%	34	10,19%
5	19,28%	20	14,41%	35	10,18%
6	19,06%	21	14,08%	36	10,17%
7	19,07%	22	13,86%	37	10,22%
8	18,91%	23	13,50%	38	10,24%
9	18,52%	24	12,77%	39	10,22%
10	18,48%	25	12,39%	40	10,19%
11	18,24%	26	12,17%	41	10,12%
12	17,72%	27	11,99%	42	10,07%
13	17,49%	28	11,61%	43	10,12%
14	16,98%	29	11,26%	44	10,10%
15	16,40%	30	11,17%	45	10,20%

Tabla 4.22: Error de clasificación para la técnica de construcción basada en camino de mínimo error mediante una matriz de construcción para la estrategia *Promedio Total* para la colección *10Newsgroups*

#Clasif	Error Clasif	#Clasif	Error Clasif	#Clasif	Error Clasif
1	79,92%	16	11,42%	31	6,51%
2	64,33%	17	11,33%	32	5,58%
3	49,80%	18	10,41%	33	5,53%
4	33,79%	19	10,38%	34	5,51%
5	20,63%	20	10,35%	35	5,46%
6	20,27%	21	8,41%	36	5,35%
7	20,16%	22	7,23%	37	5,27%
8	19,91%	23	7,05%	38	5,23%
9	19,07%	24	7,10%	39	5,19%
10	18,02%	25	7,04%	40	5,15%
11	17,09%	26	6,97%	41	5,14%
12	16,17%	27	6,74%	42	5,14%
13	15,14%	28	6,68%	43	5,08%
14	13,72%	29	6,78%	44	5,08%
15	11,95%	30	6,74%	45	5,08%

Tabla 4.23: Error de clasificación para la técnica de construcción basada en camino de mínimo error mediante una matriz de construcción para la estrategia *Promedio Total* para la colección *USPS*

Para el primer conjunto, el mínimo error obtenido es del 10.07% para 42 clasificadores, pero podemos reducir este número hasta los 24 ya que el error sólo se ha incrementado en un 2.7%. En el caso del conjunto *USPS*, el mínimo error se produce para 43 clasificadores y es del 5.08% y también podríamos reducir el número de clasificadores hasta 21 ya que el error se incrementa en poco más del 3%. A la vista de estos resultados, podríamos decir que sería posible reducir el número de clasificadores que debemos entrenar sin reducir considerablemente su eficiencia pero sí su coste computacional y temporal.

A continuación se presentan las tablas de la medida F_1 para esta estrategia de construcción y para ambas colecciones de datos:

#Clasif	Grp1	Grp2	Grp3	Grp4	Grp5	Grp6	Grp7	Grp8	Grp9	Grp10
1-vs-All (simulación)	90,10%	93,07%	93,47%	94,00%	84,06%	90,29%	87,63%	88,54%	87,63%	94,12%
US-MSVM (artículo)	87,5%	95,4%	81,2%	90,7%	85,0%	88,5%	73,0%	93,3%	80,2%	87,2%
1	0,00%	0,00%	0,00%	0,00%	0,00%	32,33%	0,00%	31,67%	0,00%	0,00%
2	0,00%	0,00%	43,81%	0,00%	0,00%	61,02%	0,00%	53,29%	51,91%	0,00%
3	0,00%	80,70%	55,02%	66,43%	0,00%	76,72%	0,00%	66,67%	60,93%	0,00%
4	0,00%	86,26%	69,17%	80,89%	65,29%	82,24%	0,00%	78,82%	76,62%	83,72%
5	83,42%	89,00%	79,80%	89,55%	81,03%	86,87%	75,12%	84,32%	77,66%	88,66%
6	84,00%	89,00%	82,00%	89,22%	79,32%	86,83%	78,95%	83,16%	77,78%	88,66%
7	83,17%	87,68%	82,18%	88,35%	75,20%	86,83%	78,53%	77,19%	77,71%	88,21%
8	83,17%	86,83%	80,77%	88,35%	74,90%	86,96%	79,78%	77,11%	77,71%	89,34%
9	83,17%	86,41%	88,42%	88,78%	75,50%	88,04%	84,66%	80,23%	78,16%	88,24%
10	83,06%	84,36%	88,42%	88,35%	71,97%	86,79%	84,66%	80,70%	76,02%	88,67%
11	81,52%	84,76%	88,42%	88,35%	72,52%	86,79%	84,66%	81,14%	76,47%	88,12%
12	81,52%	85,71%	88,42%	88,35%	72,80%	87,32%	85,26%	81,61%	78,16%	88,89%
13	82,16%	89,34%	88,42%	88,35%	70,90%	87,44%	84,82%	82,29%	78,86%	88,89%
14	82,16%	89,34%	88,54%	90,36%	70,59%	87,85%	85,57%	82,29%	80,00%	89,45%
15	82,61%	92,15%	89,12%	90,82%	69,82%	87,44%	86,15%	83,62%	79,31%	89,00%
16	83,60%	91,58%	91,28%	90,36%	70,85%	89,10%	87,31%	83,62%	79,31%	88,44%
17	84,21%	91,10%	91,28%	90,36%	71,11%	89,10%	87,00%	84,44%	79,31%	90,63%
18	84,21%	91,19%	89,58%	89,80%	70,59%	85,99%	86,43%	83,33%	79,10%	91,75%
19	84,21%	91,19%	90,16%	91,19%	70,85%	85,99%	87,56%	83,98%	79,10%	91,75%
20	85,44%	92,23%	90,16%	89,90%	81,59%	85,31%	87,13%	86,46%	85,71%	91,75%
21	85,71%	92,23%	90,16%	90,36%	84,73%	84,91%	87,13%	87,76%	86,96%	91,75%
22	86,87%	91,84%	90,16%	89,90%	84,58%	84,11%	87,13%	87,76%	85,58%	91,75%
23	86,87%	92,31%	90,00%	90,36%	85,58%	85,71%	86,87%	87,76%	86,27%	91,75%
24	86,00%	91,84%	90,00%	90,36%	85,99%	85,31%	86,87%	87,76%	86,14%	91,19%
25	86,00%	91,84%	89,11%	90,36%	85,99%	84,91%	86,46%	88,21%	86,14%	93,40%
26	86,00%	91,37%	89,00%	90,36%	85,99%	84,91%	87,63%	88,21%	86,57%	93,40%
27	86,00%	91,37%	88,67%	91,37%	85,99%	85,31%	88,08%	88,66%	86,57%	93,40%
28	86,00%	91,37%	91,28%	91,37%	85,99%	84,11%	89,23%	88,66%	86,57%	94,00%
29	86,70%	91,92%	90,72%	91,37%	86,12%	86,67%	89,69%	88,66%	86,57%	95,00%
30	87,68%	92,46%	90,72%	91,84%	86,12%	86,67%	89,12%	88,66%	86,14%	95,00%
31	88,12%	92,93%	91,84%	92,39%	86,54%	89,11%	90,45%	88,21%	86,14%	94,53%
32	88,12%	91,54%	92,31%	92,39%	86,41%	89,11%	90,45%	88,21%	86,14%	94,53%
33	87,25%	91,54%	92,31%	92,39%	87,38%	89,11%	90,45%	88,08%	86,14%	94,53%
34	88,12%	91,54%	91,75%	93,47%	86,83%	88,78%	90,91%	88,08%	86,14%	94,53%
35	87,68%	90,55%	92,31%	92,93%	86,41%	88,35%	91,37%	88,08%	85,57%	95,00%
36	87,25%	91,54%	92,31%	92,93%	85,99%	88,35%	91,37%	88,08%	85,43%	95,00%
37	87,00%	91,54%	92,31%	92,54%	85,31%	88,24%	91,37%	88,08%	85,86%	95,00%
38	87,44%	90,20%	92,31%	92,54%	85,71%	88,12%	90,00%	88,54%	85,28%	95,00%
39	87,44%	90,64%	92,31%	92,54%	85,71%	89,66%	90,45%	88,08%	85,28%	95,00%
40	87,44%	90,29%	91,84%	92,54%	86,12%	89,66%	90,45%	88,54%	84,69%	95,48%
41	87,44%	90,29%	91,84%	92,54%	86,12%	89,66%	90,45%	88,54%	84,69%	95,48%
42	87,44%	90,29%	93,88%	92,54%	86,12%	90,10%	90,00%	88,54%	84,69%	95,48%
43	87,44%	90,29%	93,33%	92,54%	85,58%	89,76%	90,00%	89,01%	84,69%	95,48%
44	87,44%	90,29%	93,33%	92,54%	85,58%	89,76%	90,45%	89,01%	85,13%	95,52%

45										
Pairwise (simulación)	87,44%	90,29%	93,33%	92,54%	85,58%	89,76%	90,00%	89,01%	85,13%	95,00%

Tabla 4.23: Medida F_1 para la técnica de construcción basada en camino de mínimo error basada en una matriz de construcción para la estrategia *Promedio Total* para *10Newsgroups*

#Clasif	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
1-vs-All (simulación)	97,25%	97,69%	92,27%	93,54%	92,38%	92,35%	95,52%	95,83%	93,62%	95,48%
US-MSVM (artículo)	93,5%	95,8%	88,0%	89,1%	93,1%	86,9%	95,5%	91,7%	90,3%	94,2%
1	0,00%	95,70%	20,48%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
2	0,00%	96,69%	26,72%	0,00%	0,00%	0,00%	85,97%	0,00%	0,00%	67,49%
3	0,00%	85,03%	32,41%	0,00%	62,55%	75,24%	86,75%	0,00%	0,00%	52,72%
4	92,35%	85,03%	46,01%	0,00%	67,52%	75,08%	88,34%	76,45%	0,00%	50,00%
5	92,35%	83,77%	58,93%	80,82%	68,53%	79,17%	88,62%	77,31%	76,39%	50,21%
6	92,35%	84,53%	58,75%	80,82%	67,95%	79,17%	88,89%	77,97%	76,16%	51,45%
7	92,35%	84,53%	58,39%	79,30%	68,09%	78,35%	88,89%	78,19%	76,16%	52,67%
8	92,35%	84,53%	56,97%	81,38%	68,24%	73,13%	88,89%	77,75%	75,89%	53,50%
9	92,35%	84,78%	56,97%	81,38%	68,24%	73,13%	88,89%	76,45%	75,62%	48,51%
10	92,35%	94,84%	55,24%	81,38%	77,68%	73,13%	88,89%	68,83%	76,66%	58,17%
11	92,84%	94,84%	55,57%	81,38%	77,68%	73,13%	88,89%	68,83%	76,66%	58,73%
12	92,84%	94,84%	55,00%	82,19%	77,68%	72,87%	88,34%	68,83%	76,92%	58,73%
13	92,84%	94,84%	54,36%	82,11%	77,68%	75,19%	88,62%	68,83%	76,92%	58,73%
14	92,84%	95,46%	54,68%	82,11%	79,66%	75,19%	88,62%	69,35%	78,32%	58,73%
15	92,84%	96,09%	53,28%	82,11%	80,67%	75,19%	88,62%	91,30%	79,17%	93,84%
16	92,84%	96,48%	53,66%	83,74%	81,56%	75,19%	88,62%	91,30%	79,17%	93,84%
17	93,15%	96,48%	53,74%	83,74%	81,56%	75,68%	89,78%	91,30%	79,17%	93,84%
18	93,15%	96,48%	53,51%	83,74%	81,56%	72,44%	89,78%	92,09%	79,17%	93,84%
19	93,47%	96,48%	53,81%	83,74%	81,56%	72,66%	89,78%	92,09%	79,17%	93,84%
20	93,47%	96,30%	53,81%	83,74%	81,23%	72,66%	89,78%	92,09%	79,17%	93,84%
21	97,10%	96,30%	86,55%	92,83%	91,04%	91,29%	93,13%	94,77%	92,49%	94,97%
22	96,95%	96,30%	86,91%	93,83%	90,82%	91,29%	92,81%	94,44%	92,49%	95,26%
23	97,36%	96,30%	87,13%	93,83%	91,09%	91,57%	94,08%	94,08%	91,62%	95,26%
24	97,21%	96,30%	87,32%	93,83%	90,32%	92,12%	94,08%	95,14%	90,96%	95,56%
25	97,21%	96,30%	88,67%	93,83%	91,13%	91,84%	93,81%	95,14%	92,73%	95,29%
26	97,49%	96,48%	90,86%	93,83%	92,35%	91,84%	93,53%	95,10%	93,05%	95,58%
27	97,49%	96,48%	90,86%	93,83%	92,80%	91,84%	93,53%	95,14%	93,05%	95,58%
28	97,63%	96,48%	90,86%	93,29%	92,80%	92,07%	93,84%	95,14%	92,73%	95,58%
29	97,63%	96,48%	90,86%	93,58%	92,35%	92,07%	93,84%	95,47%	93,05%	95,84%
30	97,63%	96,30%	91,09%	93,01%	92,08%	92,07%	93,84%	95,47%	92,73%	95,84%
31	97,63%	96,69%	91,32%	93,29%	92,35%	92,07%	93,57%	95,47%	93,66%	95,56%
32	97,63%	96,69%	91,32%	93,29%	93,30%	92,35%	93,57%	95,86%	93,66%	95,56%
33	97,49%	96,89%	91,32%	93,29%	92,80%	92,35%	93,53%	95,86%	93,66%	95,56%
34	97,49%	96,89%	91,04%	93,29%	92,57%	92,07%	93,81%	95,86%	93,37%	95,26%
35	97,49%	96,50%	91,04%	93,29%	92,57%	92,35%	93,81%	95,53%	93,37%	95,26%
36	97,63%	97,10%	91,04%	93,29%	92,57%	92,35%	94,40%	95,86%	93,66%	95,26%
37	97,63%	96,90%	91,04%	93,29%	92,57%	92,35%	94,40%	95,53%	93,66%	95,26%
38	97,62%	96,90%	90,64%	93,58%	92,57%	92,64%	94,40%	95,53%	93,66%	95,26%
39	97,62%	97,50%	90,64%	93,58%	92,57%	92,64%	94,40%	96,53%	93,66%	95,26%
40	97,90%	97,50%	90,86%	93,58%	92,57%	92,64%	94,08%	96,53%	93,09%	95,26%
41	98,05%	97,50%	91,32%	93,58%	92,57%	92,64%	94,36%	96,53%	93,09%	95,26%
42	98,04%	97,50%	91,09%	93,58%	92,57%	92,64%	94,67%	96,53%	93,09%	95,26%
43	98,04%	97,50%	91,32%	93,58%	92,57%	92,64%	94,40%	96,53%	93,09%	95,26%
44	98,06%	97,50%	91,04%	93,58%	92,57%	92,64%	94,96%	96,53%	93,37%	95,26%
45	98,06%	97,50%	91,04%	93,58%	92,57%	92,64%	94,96%	96,53%	93,37%	95,26%
Pairwise (simulación)	98,20%	97,89%	91,41%	93,29%	93,33%	92,64%	94,40%	96,53%	92,73%	95,56%

Tabla 4.24: Medida F_1 para la técnica de construcción basada en camino de mínimo error basada en una matriz de construcción para la estrategia *Promedio Total* para *USPSs*

4.5.6. Comparación de los métodos Deconstructivos y Constructivos

Se ha visto en el análisis particular de los métodos de construcción y de deconstrucción, que la técnica basada en el camino de error mínimo basada en la estrategia de predicción de *Promedio Total* es con la cual se han obtenido los mejores resultados para las dos colecciones de datos bajo estudio.

Para sendos conjuntos, se va a comparar de manera gráfica los resultados logrados con esta técnica de predicción o combinación de clasificadores, con los obtenidos en los casos de clasificación *1-vs-All* y las basadas en la estrategia descrita por los autores, US-MSVM basada en *PPS*, para ambas técnicas de construcción y deconstrucción.

Se puede observar gráficamente que para ambos conjuntos los métodos constructivos son subóptimos con respecto a los métodos equivalentes que utilizan la misma técnica. Esto es debido a que los métodos deconstructivos tienen información a priori de las prestaciones de todos los clasificadores individuales pareados ya que han sido entrenados anteriormente. De esta manera y al utilizar un algoritmo de optimización “greedy”, podemos escoger aquellas combinaciones de clasificadores que no alteren demasiado el resultado final del clasificador combinado buscando minimizar el error de clasificación en cada paso.

No obstante, aunque los métodos constructivos sean subóptimos con respecto a los deconstructivos con ellos se puede reducir la carga computacional y temporal de la clasificación ya que solamente se entrenará el número necesario de clasificadores pareados para conseguir unas buenas prestaciones, en nuestro caso un valor del error de clasificación aceptable, del clasificador combinado final. De los dos métodos de construcción utilizados escogeríamos el que hemos propuesto nosotros en este proyecto basado en la búsqueda de un camino de error mínimo con la estrategia de combinación que usa la medida de confianza *Promedio Total* ya que se reduce el error de clasificación en todo el rango de clasificadores con respecto al método utilizado por los autores en el artículo basada en una técnica de muestreo de incertidumbre mediante la medida *PPS*.

Para concluir, hemos comprobado que con la técnica de construcción que hemos propuesto hemos conseguido resultados aceptables reduciendo el número de clasificadores que hay que entrenar entorno a la mitad y de este hecho ha conllevado la reducción del coste computacional que estábamos buscando.

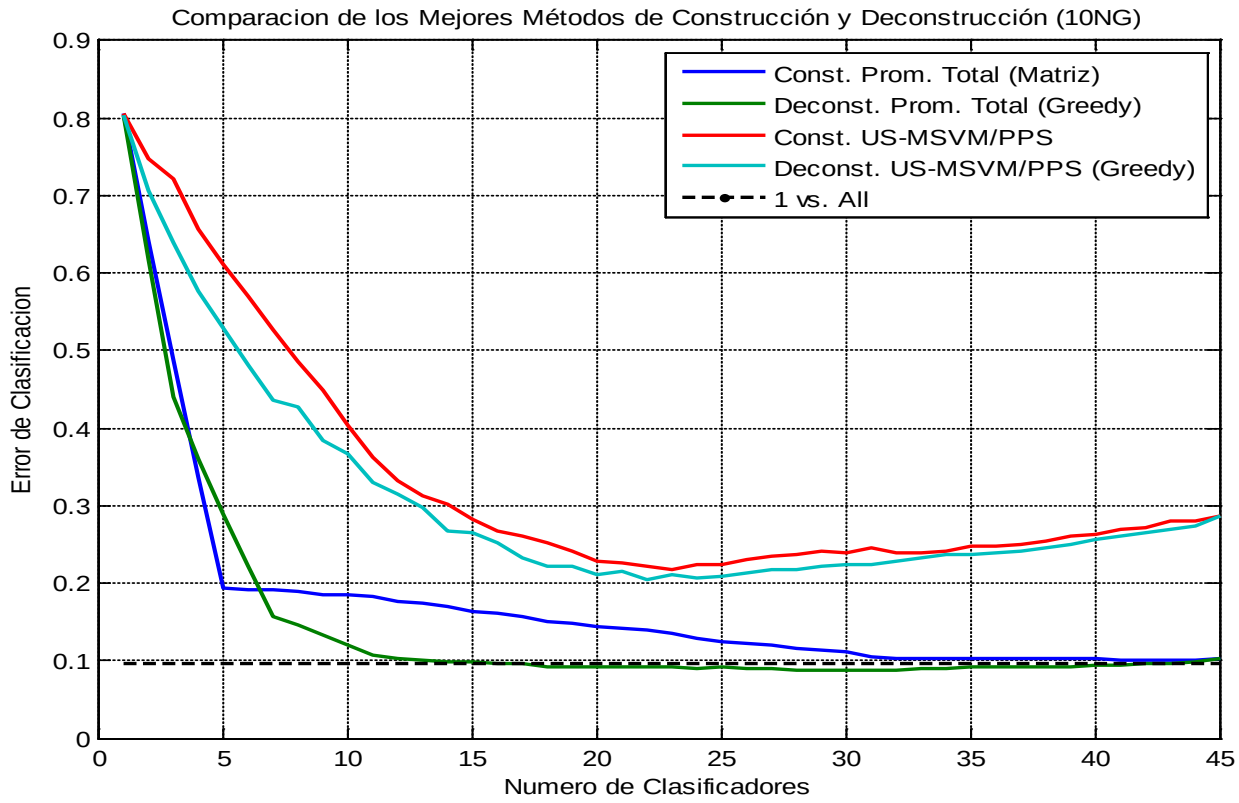


Figura 4.21: Comparación de la evolución del error de clasificación en función del número de clasificadores de las técnicas de construcción y deconstrucción y las mejores estrategias de predicción para *10Newsgroups*

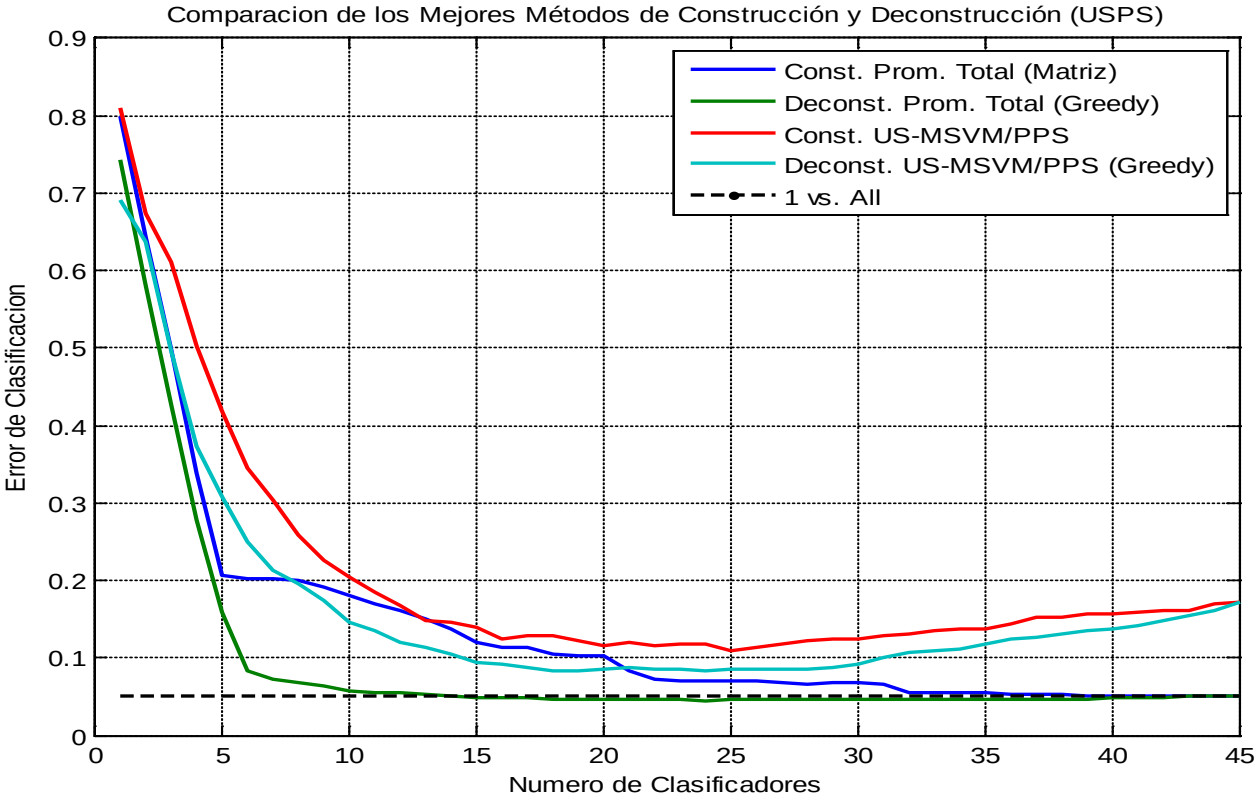


Figura 4.22: Comparación de la evolución del error de clasificación en función del número de clasificadores de las técnicas de construcción y deconstrucción y las mejores estrategias de predicción para *USPS*

Capítulo 5

Conclusiones y Líneas Futuras de Trabajo.

5.1. Conclusiones

Este proyecto se ha centrado en el estudio de dos aproximaciones para problemas de clasificación en entornos multiclase basadas en la combinación de clasificadores SVM pareados con las que se ha buscado la reducción del coste temporal y computacional de la evaluación con las técnicas clásicas. Igualmente se han podido evaluar el buen comportamiento y las prestaciones de las mismas atendiendo a la eficacia y precisión en la clasificación.

Las dos aproximaciones que se han realizado se basan en la combinación de diversos clasificadores SVM binarios pareados con las que se consigue predecir la categoría o clase de todas las muestras de las colecciones de datos de problema. La primera que se ha analizado es una aproximación deconstructiva basada en la poda de clasificadores y la segunda es una aproximación constructiva. Así mismo la combinación de los clasificadores binarios para la predicción de la categoría de cada patrón de muestra se ha realizado mediante varias estrategias como por voto mayoritario y reglas u operadores básicos estadísticos.

En el capítulo anterior hemos presentado los resultados experimentales de diversas estrategias de clasificación y ahora se van a resumir las conclusiones a las que hemos podido llegar tras nuestras investigaciones.

En primer lugar se hizo una simulación del experimento US-MSVM basado en muestreo de incertidumbre propuesto por los autores en el artículo. Para realizar nuestros experimentos tuvimos en cuenta todas las indicaciones y restricciones expuestas en él. Dentro de estas indicaciones podemos nombrar el procesado de las colecciones de datos, donde se redujo la colección *20Newsgroups* a un conjunto de datos más pequeño formado por las 10 categorías mostradas en la Tabla 4.1 y se crearon del mismo modo que ellos comentan para ambas colecciones los conjuntos de entrenamiento y test con los que íbamos a trabajar. Otras restricciones fueron tomadas del artículo para realizar las simulaciones de los experimentos bajo las mismas condiciones, como los parámetros de los

clasificadores SVM o del método US-MSVM. Bajo este entorno de simulación se realizaron los experimentos del algoritmo pero no se consiguieron replicar los resultados publicados para ninguno de los conjuntos de prueba siendo los resultados obtenidos mucho peores que los publicados, llegando a tener diferencias incluso del 50%. Por esta razón y vistas las grandes diferencias de resultados, pensamos en realizar nuevos métodos o aproximaciones para SVM multiclase con los que sí se obtuvieran buenas prestaciones.

De este modo, se plantearon varias aproximaciones para problemas de multiclasificación basadas en la combinación de clasificadores SVM pareados en las que se utilizaban también diversas estrategias de combinación o predicción de la clase final. Para ambas colecciones de datos bajo estudio, se realizaron varios experimentos en base a las dos aproximaciones multiclase y a las técnicas de combinación propuestas, estudiando las prestaciones que se obtienen desde la técnica más simple o "baseline" a otras ya un poco más elaboradas como las basadas en algoritmos de optimización "greedy". Posteriormente se ha hecho una comparación de resultados y prestaciones de todas las técnicas utilizadas.

Así, en un principio, se estudiaron las prestaciones de la técnica deconstructiva basada en poda de clasificadores pareados en la que se aplicaron diferentes métodos de combinación de dichos clasificadores, desde métodos basados en voto por mayoría hasta métodos de medidas de niveles de confianza basados en la probabilidad de que un clasificador determine que un patrón pertenezca a una categoría u otra. Se comprobó que, para ambos conjuntos, la técnica de eliminación de clasificadores binarios que mejores resultados daba era la basada en un algoritmo "greedy" para encontrar un camino de mínimo error que utilizaba una estrategia de combinación basada en la regla *Promedio Total*. Con ésta incluso se logra reducir el error de clasificación obtenido con las técnicas de combinación clásicas *Pairwise* y *1-vs-All*, para un número de clasificadores mucho menor (un tercio) que los que hay que utilizar en la primera técnica y no mucho mayor (5 más) para la segunda técnica. Para este apartado, se puede concluir que con los métodos reconstructivos realmente no reducimos el coste computacional ya que necesitamos tener previamente entrenados todos los clasificadores pareados, pero realmente nos sirven para comprobar que sería posible, mediante unos métodos constructivos equivalentes, reducir dicho número sin perder muchas prestaciones pero sí consiguiendo una reducción de la carga computacional.

De esta manera, se diseñaron varios métodos constructivos en los que se iban añadiendo clasificadores pareados siguiendo las mismas técnicas que se utilizaban en los métodos deconstructivos para la eliminación de los mismos. Así mismo, se aplicaron en cada uno de ellos las mismas estrategias de combinación de clasificadores binarios SVM que en aquellos. Se comprobó, para sendas colecciones, que el método de adición de clasificadores que mejores resultados daba y al mismo tiempo reducía la carga computacional era el equivalente al método deconstructivo basado en encontrar un camino "greedy" de mínimo error que utilizaba una estrategia de predicción basada en *Promedio Total*. En este método de construcción se elegían en cada paso los siguientes clasificadores pareados que había que entrenar mediante la creación de una matriz cuyos elementos representaban el error acumulado de cada par de clases hasta ese momento. Se observó experimentalmente que, para ambos conjuntos de datos, se podrían reducir el número de clasificadores necesarios que habría que entrenar a la mitad con respecto a la técnica de combinación de clasificadores tradicional *Pairwise* y no aumenta significativamente con respecto a los utilizados en *1-vs-All* .

Para concluir, podemos decir que con la técnica constructiva mencionada en el párrafo anterior conseguimos reducir notablemente la carga computacional que conllevaban estas las técnicas clásicas de combinación de clasificadores pareados *1-vs-All* y *Pairwise*. Con respecto a la primera de ellas, aunque tenemos que entrenar mayor número de clasificadores estos lo harán con un conjunto de menor tamaño, es decir, con menor número de muestras por lo que el coste del entrenamiento en base al tiempo de cómputo y la complejidad se reducirá. Por otra parte, está claro que reducimos la carga computacional en comparación con la técnica *Pairwise* ya que será necesario entrenar un número mucho menor de clasificadores pareados para conseguir unos resultados aceptables sin perder casi eficiencia o prestaciones.

5.1.1. Reducción de la carga y el tiempo de cómputo

Para concluir se va a presentar un breve resumen de los tiempos de cómputo que requieren los diferentes métodos de clasificación para ambas colecciones de datos. Por tanto, se van a comparar desde las técnicas clásicas o el método utilizado en el artículo hasta el más interesante y representativo de los propuestos en este proyecto.

		Tiempo de Cómputo en Entrenamiento		**Tiempo de Cómputo en Test**		**Tiempo de Cómputo Total**
Técnica de Clasificación	**Nº de clasificadores pareados necesarios**	**Para un clasificador pareado**	**Para todos los clasif. pareados**	**Para un clasificador pareado**	**Para todos los clasif. pareados**	**Para todos los clasif. pareados**
1-vs-All	10	32 seg.	5 min.	2 seg.	30 seg.	5.5 min.
Pairwise	45	3 seg.	2.25 min.	1 seg.	45 seg.	3 min.
US-MSVM (PPS)	10	3 seg.	30 seg.	~ 0 seg.	~ 0 seg.	30 seg.
Matriz de Construcción + Promedio Total	24	3 seg.	1.5 min.	~ 0 seg.	~ 0 seg.	1.5 min.

Tabla 5.1: Tiempos de cómputo total, de entrenamiento y test para diferentes técnicas de clasificación para la colección *10Newsgroups*

		Tiempo de Cómputo en Entrenamiento		**Tiempo de Cómputo en Test**		**Tiempo de Cómputo Total**
Técnica de Clasificación	**Nº de clasificadores pareados necesarios**	**Para un clasificador pareado**	**Para todos los clasif. pareados**	**Para un clasificador pareado**	**Para todos los clasif. pareados**	**Para todos los clasif. pareados**
1-vs-All	10	4.6 min.	45 min.	15 seg.	2.5 min.	48 min.
Pairwise	45	15 seg.	11 min.	5 seg.	4 min.	15 min.
US-MSVM (PPS)	16	25 seg.	7 min.	~ 0 seg.	~ 0 seg.	7 min.
Matriz de Construcción + Promedio Total	23	25 seg.	10 min.	~ 0 seg.	~ 0 seg.	10 min.

Tabla 5.2: Tiempos de cómputo total, de entrenamiento y test para diferentes técnicas de clasificación para la colección *USPS*

Podemos observar en las tablas anteriores que son las técnicas tradicionales de clasificación son las que mayor tiempo de cómputo requieren para ambas colecciones de datos, en cambio, es el método propuesto en el artículo que utiliza la medida PPS el que menor tiempo emplea. Si comparamos estos resultados con los que se obtienen con el método constructivo con la técnica de adicción de clasificadores pareados basada en una matriz de construcción que utiliza la estrategia de combinación *Promedio Total*, que es con el que mejores resultados hemos obtenido, podemos ver que no aumenta notablemente el tiempo empleado con respecto a US-MSVM y si se reduce de manera significativa con respecto a la técnica *1-vs-All* (casi en 5 veces).

Por tanto, con el método propuesto por nosotros en este proyecto conseguimos unos buenos resultados con respecto a la eficiencia de la clasificación, es decir, logramos errores aceptables al entrenar solamente la mitad de los clasificadores pareados necesarios con respecto a *Pairwise* y pocos más que con *1-vs-All*, sin aumentar mucho su valor. En este apartado hemos visto que también conseguimos una reducción del tiempo total de cómputo de la clasificación con respecto a estas dos técnicas tradicionales por lo que hemos cumplido el objetivo marcado en el proyecto de reducir la carga computacional en técnicas de clasificación multiclase.

5.2. Líneas de Trabajo Futuras

En esta última sección se van a presentar dos posibles líneas de trabajo para el futuro con las que probablemente se pueda reducir la carga computacional en entornos de clasificación multiclase.

5.2.1. Generalización del modelo

Para hacer una generalización del modelo y de los resultados proponemos utilizar para realizar los experimentos tres conjuntos de datos añadiendo a los de entrenamiento y test, un nuevo conjunto de validación.

Este conjunto de validación se usaría para elegir el mínimo número de clasificadores pareados que habría que entrenar y con el que se conseguiría unas buenas prestaciones así como una reducción significativa de la carga temporal y computacional de la clasificación. Entonces se entrenaría solamente el número de clasificadores fijado por el conjunto de validación evaluándose el error de clasificación utilizando el conjunto de test, y de esta manera se haría una generalización del problema.

En este proyecto somos conscientes de que hay ocasiones en que se debería haber utilizado el conjunto de validación para ser más justos al comparar los resultados de los experimentos realizados con los obtenidos con las técnicas tradicionales de clasificación como *Pairwise* o *1-vs-All*. A parte de este hecho, los resultados obtenidos son válidos ya que a nosotros nos interesa y nos permiten comparar la evolución de los errores de clasificación con el número de clasificadores entrenados de todos los experimentos de todos los métodos propuestos ya que en todos ellos se ha utilizado el mismo conjunto para medir sus respectivas prestaciones.

5.2.2. Mejora del método US-MSVM

Una de las posibles líneas de estudio futuras sería continuar con la idea propuesta en el artículo [Ye y Shang-Teng, 2007] y seguir utilizando una técnica basada en muestreo de incertidumbre pero cambiando algunas de las medidas utilizadas en el mismo.

Por ejemplo, la utilización de la medida PPS quizá no sea la más adecuada para luego medir distancias o similitudes entre las clases y por ello sería conveniente hacer una especie de codificación que transforme los valores de las PPS a 0, +1 y -1. Serían +1 si la PPS supera un cierto umbral (por ejemplo 0.7) lo que significa que es muy probable que ese patrón pertenezca a la clase etiquetada como “1” y del mismo modo sería -1 si la PPS es menor que una cota (por ejemplo 0.3) lo que significa que es muy probable que pertenezca a la clase “-1”. En este sentido todos los valores que se encuentren entre estos umbrales se codificarán con un 0 o “indefinido” ya que no tiene una alta posibilidad de ser de ninguna de las dos clases. Tras esta codificación se calcularía la medida de la incertidumbre utilizando en vez de la distancia euclídea otra como la de Hamming.

En este sentido es probable que mejoren las prestaciones del método propuesto basado en muestreo de incertidumbre e incluso pueda reducirse el número de subclasificadores pareados que es necesario entrenar.

5.2.3. Uso de Técnicas de Aprendizaje Semi-supervisadas para SVM

Otra de las posibles líneas para el trabajo futuro es utilizar en vez de técnicas de aprendizaje supervisado algún método de clasificación semi-supervisadas.

Este tipo de algoritmos utilizan no sólo el conjunto de datos etiquetados sino también los datos no etiquetados para generar un modelo de clasificación y predicción adecuado. Este modelo se entrena con el conjunto etiquetado y más tarde se asigna una determinada clase a los datos no etiquetados mediante técnicas de agrupamiento.

Tienen como ventajas que se logra una mayor exactitud en la clasificación con respecto a los métodos supervisados o no-supervisados y tampoco se necesita de un gran esfuerzo humano dado que no es necesario etiquetar todo el conjunto de datos.

Estas técnicas de aprendizaje podrían ser muy útiles para reducir el número de clasificadores que deben ser entrenados y este hecho reportaría indudablemente una reducción de la carga o coste computacional de la clasificación.

Para más información acerca como las técnicas de aprendizaje semi-supervisadas podrían beneficiar a la reducción de la carga computacional véase en el apéndice A. Del mismo modo puede encontrarse una introducción a estas técnicas basadas en máquinas de vectores soporte y su aplicación a algunos problemas de clasificación como los que hemos tratado en el apéndice B.

Apéndice A

Aprendizaje Semi-supervisado para la posible Reducción de la Carga Computacional

En un problema en la que existen múltiples clases si utilizamos como la estrategia de combinación de varios problemas de clasificación binarios *Pairwise* o *1-vs-1* para realizar una aproximación multiclase, con una técnica de aprendizaje supervisada se deben entrenar tantos clasificadores como pares de clases.

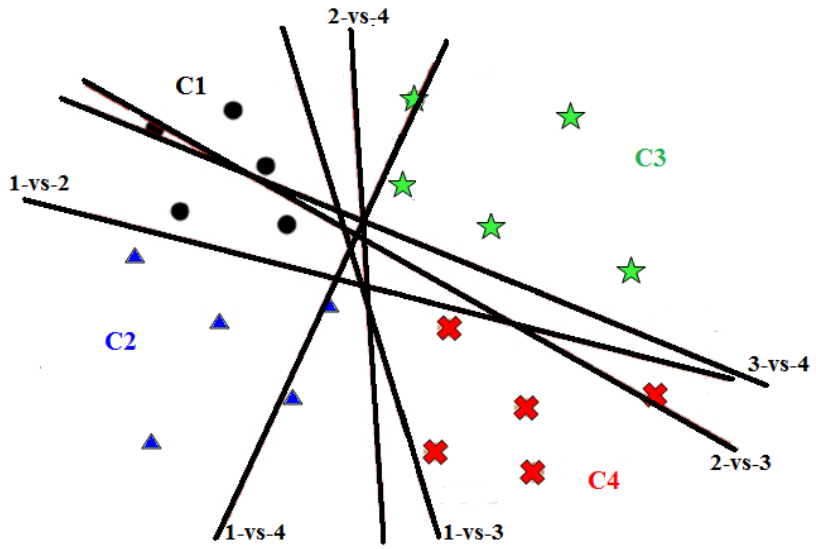


Figura A.1: Ejemplo de fronteras para clasificación *Pairwise* basada en una técnica de aprendizaje supervisado para un problema con 4 clases. Se necesitan entrenar 6 clasificadores.

Si se utiliza una técnica de aprendizaje semi-supervisada al entrenar un clasificador formado por dos determinadas clases no solamente deben tenerse en cuenta los datos pertenecientes a esas dos clases sino que también se deben tener en cuenta también el resto de ellas como muestras no etiquetadas a las que se les asociará una determinada categoría mediante algún algoritmo de agrupamiento.

De esta manera sería interesante realizar un estudio basado en estas técnicas para entornos de clasificación multiclase para comprobar si es posible reducir el número de clasificadores

que es necesario entrenar. La idea es que quizá con el uso de estas técnicas semi-supervisadas sería suficiente entrenar un determinado número de clasificadores con el que se consiga separar las clases existentes en el problema.

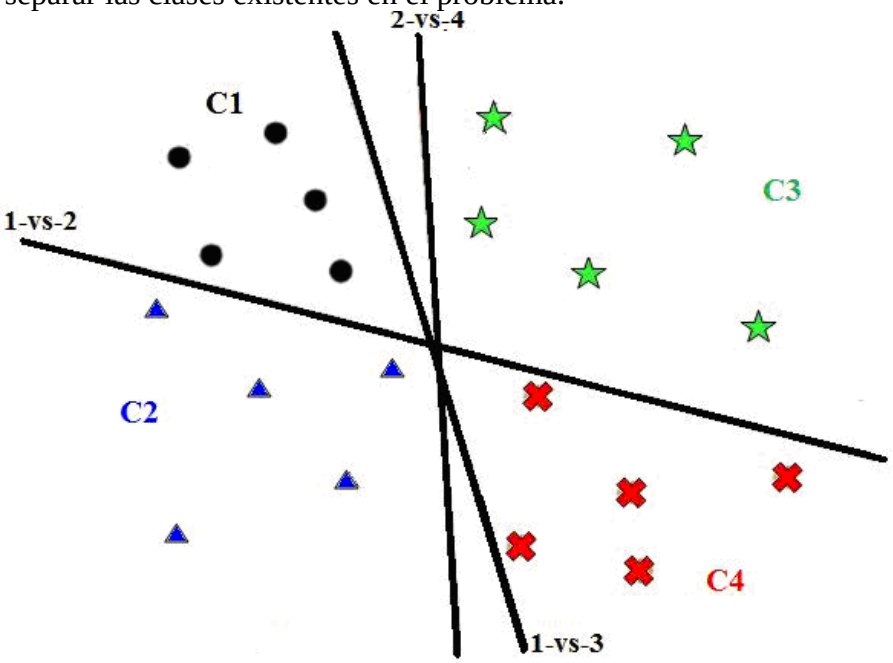


Figura A.2: Hipótesis de fronteras para clasificación basada en una técnica de aprendizaje semi-supervisado para un problema con 4 clases. Quizá solamente sería necesario entrenar sólo 3 clasificadores para separar todas las clases.

Ahora se presentan algunas de las ventajas que podrían darse al utilizar las técnicas semi-supervisadas. Primero, podría reducirse la carga computacional si se pudiera comprobar el hecho comentado anteriormente de que con estos métodos se puede disminuir el número de clasificadores que hay que entrenar.

Por otro lado, al entrenar un clasificador binario formado por dos determinadas clases el resto de las muestras que no tienen asociadas inicialmente una categoría se reparten y etiquetan como una de esas dos clases, de este modo se posee una información más precisa ya que al final se conoce a que categoría pertenecen todas las muestras. También, de esta manera, se reduce el número de muestras por clasificador con respecto al caso *1-vs-All* y no aumenta en gran medida con respecto al caso *Pairwise* por lo que también se tiene esa ventaja de esta técnica clásica.

Por último presentamos una de las principales desventajas de estas técnicas que es que presenta un alto coste de entrenamiento.

Apéndice B

Introducción a la Técnicas de Aprendizaje Semi-supervisado para SVM

Se podría estudiar el comportamiento, las prestaciones y la bondad de algunos de los algoritmos semi-supervisados utilizados más comúnmente en el área de la clasificación así como si realmente ayudaría en la reducción de clasificadores pareados y, por tanto, en la carga computacional. Los más habituales son los basados en EM, Self-training o Bootstrapping, Co-training y Máquinas de Soporte Vectorial transductivo o semi-supervisado(T-SVM o S^3VM), etc.

Nosotros proponemos utilizar el algoritmo de aprendizaje basado en Máquinas de Vectores Soporte Semi-supervisado para facilitar la adaptación desde nuestros experimentos ya que es la extensión natural de la técnica SVM utilizada. A continuación se presenta esta técnica de clasificación.

B.1. Máquinas de Vectores Soporte Semi-Supervisadas o Transductivas (S3VM o TSVM)

Esta técnica de aprendizaje es una extensión del método supervisado basado en máquinas de vector soporte, SVM. Como ocurre en este modelo, se trata de encontrar un hiperplano que separa los datos del conjunto de entrenamiento maximizando el margen de separación pero, dado que es un método semi-supervisado, también se utiliza un conjunto de datos no etiquetados.

Este método se basa en poner etiquetas al conjunto de ejemplos no etiquetados que hagan máximo el margen de separación tanto para los datos etiquetados originales como para los no etiquetados al que se les acaba de asignar una etiqueta.

El objetivo de la utilización de los ejemplos no etiquetados junto con los ejemplos no etiquetados es obtener unos límites de decisión en regiones poco pobladas o de baja densidad de datos que maximicen después el margen en áreas más pobladas.

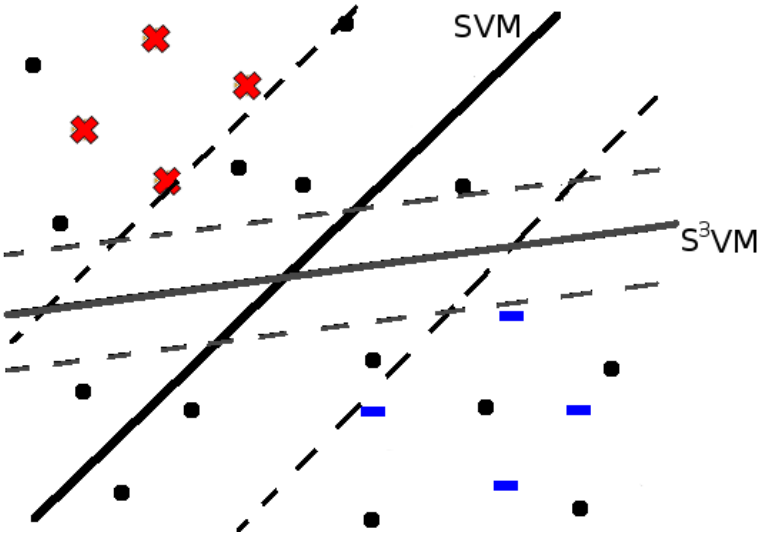


Figura B.1: Comparación de las función de clasificación para SVM vs S^3VM. Los documentos etiquetados para las dos clases están representados por x/- y los no etiquetados lo están por puntos

La función del hiperplano que hay que encontrar sigue siendo la misma que para SVM supervisado:

$$f(\underline{x}) = \underline{w}^T \cdot \underline{x} + b \tag{B.1}$$

En el caso de SVM transductivo o semi-supervisado, a la función de maximización del margen, en un caso no-separable, se le añade un nuevo término referente al conjunto de datos no etiquetado.

$$mín \quad \frac{1}{2} \cdot \| \underline{w} \|^2 + C \cdot \sum_{i=0}^{n} \xi_i^d + C^* \cdot \sum_{j=0}^{k} \xi^{*d}_j \tag{B.2}$$

Sujeto a las siguientes restricciones:

$$\begin{aligned}
&\forall_{i=0}^{n} : \; y_i \; (w \cdot x_i + b) \geq 1 - \xi_i \\
&\forall_{j=0}^{k} : \; y_j^* \; (w \cdot x_j + b) \geq 1 - \xi_j^* \\
&\forall_{i=0}^{n} : \; \xi_i > 0 \\
&\forall_{j=0}^{k} : \; \xi_j^* > 0 \\
&\forall_{j=0}^{k} : \; y_j^* \in \{-1, +1\}
\end{aligned} \tag{B.3}$$

B.2. Aproximaciones Multiclase de las Máquinas de Vectores Soporte Semi-Supervisadas

Actualmente existen varios estudios que han investigado la transformación de la técnica supervisada SVM a la técnica semi-supervisada para problemas multiclase.

En primer lugar, [Yajima y Kuo, 2006] proponen la aproximación directa que traslada la función multiclase directa al entorno semi-supervisado aunque puede resultar algo costosa por la gran cantidad de variables que se deben utilizar.

Por otro lado, algunos trabajos emplean otros enfoques para consecución de la técnica S^3VM multiclase como los de [Qi et al., 2004] que utilizan técnicas Fuzzy C-Means (FCM) para la predicción de la clase de los datos no etiquetados y para el mismo objetivo el trabajo de [Xu y Schuurmans, 2005] que se basa en técnicas de agrupamiento o clustering.

Por último, [Chapelle et al., 2003] propone el traslado de las técnicas clásicas de combinación de clasificadores binarios como *1-vs-All* y *Pairwise* o *1-vs-1* al entorno semi-supervisado. Siguiendo estas propuestas varios investigadores españoles, [Zubiaga et al., 2009], publicaron recientemente un estudio comparativo de aproximaciones a SVM multiclase semi-supervisado.

Referencias Bibliográficas.

[Bennett y Demiriz, 2002]: Bennett, K.P., Demiriz, A., Maclin, R.: *Exploiting unlabeled data in ensemble methods.* In: KDD. (2002)

[Bensaid et al., 1996]: Bensaid, A.M., Hall, L.O., Bezdek, J.C. and Clarke, L.P., *Partially supervised clustering for image segmentation,* Patter Recognition 29, pp 859 - 871, (1996).

[Blum y Chawla, 2001]: Blum, A. and Chawla, S., *Learning from labelled and unlabeled data using graph mincuts*, In Proc. 18th. Int. Conf on Machine Learning, pp 19 - 26, (2001).

[Blum y Mitchell, 1998]: Blum, A., & Mitchell, T. (1998). *Combining labeled and unlabeled data with co-training*. *COLT*: Proceedings of the Workshop on Computational Learning Theory.

[Boser et al., 1992]: B. E. Boser, I. M. Guyon, and V. Vapnik: *A training algorithm for optimal margin classifiers.* pp. 144-152, 1992

[Burges, 1998]: C. J. C. Burges: *A tutorial on support vector machines for pattern recognition,* Data Mining and Knowledge Discovery, vol. 2, pp. 121–167, 1998.

[Castelliy Cover, 1995]: Castelli, V. and Cover, T.M., *On the exponential value of labelled samples.* Pattern Recognition Letters 16, pp 105 - 111, (1995).

[Chapelle et al., 2003]: Chapelle, M. Chi and A. Zien: *A Continuation Method for Semi-supervised SVMs*. Proceedings of ICML'06, the 23rd International Conference on Machine Learning, 2003.

[Cortes y Vapnik, 1995]: C. Cortes and V. Vapnik: *Support-vector networks,* Machine Learning, pp.*273-297,* 1995

[Cristianini y Shawe-Taylor, 2000]: N. Cristanini and J. Shawe-Taylor: *An introduction to Support Vector Machines and other kernel-based learning methods,* pp. 83-122. Cambridge Univ. Press, 2000

[Dempster et al., 1977]: Dempster, A., Laird, N. and Rubin, D., *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal Statistical Society, Series B, (1997).

[Duda y Hary, 1973]: Duda, R.O. and Hart, P.E., *Pattern Classification, and Scene Analysis.* John Wiley & Sons, New York, (1973).

[Efron y Tibshirani, 1993]: B. Efron y R. J. Tibshirani. *An Introduction to the Bootstrap*. Londres: Chapman &Hall, 1993.

[Fisher, 1987]: Fisher, D.H. (1987), *Knowledge Acquisition via Incremental Conceptual Clustering*, Machine Learning 2:139-172, reprinted in Shavlik & Dietterich (eds.), Readings in Machine Learning, section 3.2.1.

[Fix y Hodges, 1951]: E. Fix, J. Hodges: *Discriminatory analysis, nonparametric discrimination: consistency properties*, Technical Report 4, Project No. 21-49-004, USAF School of Aviation Medicine, Randolph field, Texas, 1951.

[Hsu y Lin, 2002]: C.-W. Hsu and C.-J. Lin: *A comparison of methods for multi-class support vector machines*. IEEE Transactions on Neural Networks, 13:2, pp. 415-425, 2002

[Joachims, 1998]: T. Joachims: *Text categorization with support vector machines: learning with many relevant features.* pp. 137-142, 1998

[Joachims, 2001]: T. Joachims: *Learning to classify text using support vector machines: methods, theories and algorithms.* 2001

[Lewis y Gale, 1994]: D.D. Lewis and W. A. Gale: *A sequential algorithm for training text classifiers*, pp. 3–12, 1994

[Lin et al., 2007]: H.-T. Lin, C.-J. Lin and R.C. Weng: *A note on Platts's probabilistic outputs for Support Vector Machine Learning*, 68 (3), pp. 267-276, 2007.

[MacQueen, 1967]: MacQueen, J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations.* Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. **1**. University of California Press. pp. 281–297.

[Nigam y Ghani, 2000]: K. Nigam y R. Ghani. *Analyzing the effectiveness and applicability of co-training*. Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 86{93, McLean, VA, EE.UU., 2000.

[Qi et al., 2004]: H.-N. Qi, J.-G. Yang, Y.-W, Zhong and C. Deng: *Multi-class SVM Based Remote Sensing Image Classification and its Semisupervised Improvement Scheme*. Proceedings of the 3rd ICMLC, 2004.

[Rifkin y Klautau, 2004]: R. Rifkin and A. Klautau: *In Defense of One-Vs-All Classification*, ,Journal of Machine Learning Research, 5, pp. 101-141, 2004.

[Salton y Buckley, 1988]: G. Salton and C. Buckley: *Term-weighting approaches in automatic text retrieval*. Information Processing and Management, *24*(5), 513-523, 1988.

[Theodoris y Koutroumbas, 1998]: S. Theodoridis, K. Koutroumbas: *Pattern Recognition*, Academic Press, 1998.

[Weston y Watkins, 1999]: Weston y C. Watkins: *Multi-class Support Vector Machines*. Proceedings of ESAAN, the European Symposium on Artificial Neural Networks, 1999

[Xu y Schuumans, 2005]: L. Xu and D. Schuurmans: *Unsupervised and Semi-supervised Multiclass Support Vector Machines* Proceedings of AAAI'05, the 20th National Conference on Artificial Intelligence, 2005

[Yajima y Kuo, 2006]: Yajima and T.-F. Kuo: *Optimization Approaches for Semi-Supervised Multiclass Classification*. Proceedings of ICDMW'06, the 6th International Conference on Data Mining, 2006.

[Ye y Shang-Teng, 2007]: W. Ye and H.Shang-Teng, 2007. *Reducing the number of sub-classifiers for pairwise multi-category support vector machines*. Pattern Recognition Letters, v.28 n.15, p.2088-20093.

[Zubiaga et al., 2009]: A. Zubiaga, V. Fresno y R. Martínez: *Comparativa de aproximaciones a SVM Semisupervisado Multiclase para Clasificación de Páginas Web*, In revista de la Sociedad Española para el Procesamiento del Lenguaje Natural, SEPLN, Vol. 42, pp. 63-70, 2009.